UNIVERSIDAD AUTONOMA DE MADRID

Escuela Politécnica Superior

20|21

Trabajo fin de grado

Estudio de técnicas de data science para la predicción de rendimientos deportivos

Íñigo Gómez Carvajal

Escuela Politécnica Superior
Universidad Autónoma de Madrid
C\Francisco Tomás y Valiente nº 11

Campus Internacional excelencia UAM CSIC+

www.uam.es

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Estudio de técnicas de data science para la predicción de rendimientos deportivos

Autor: Íñigo Gómez Carvajal
Tutor: Lara Quijano Sánchez

junio 2021

Íñigo Gómez Carvajal
Estudio de técnicas de data science para la predicción de rendimientos deportivos

Íñigo Gómez Carvajal
C\ Eladio Lopez Vilches, 15

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

Para mis padres, de su McFly.

No es la voluntad de ganar,
sino la voluntad de prepararse para ganar
lo que marca la diferencia

Bear Bryant

Agradecimientos

En primer lugar me gustaría agradecer a mi tutora, Lara Quijano Sánchez, por su iniciativa a la hora de coger mi propuesta inicial, por su implicación, dedicación y sobre todo por mantener una ambición acorde con la mía que me ha ayudado a mantener la motivación durante tantos meses.

También me gustaría dedicar este trabajo a los amigos que he realizado a lo largo de estos 4 años de carrera: Alberto, Alejandro, Iván y Julio. Por todo el apoyo recibido y por haber sido los mejores amigos que uno podría pedir.

Pero, sin duda alguna, este trabajo se lo dedico a mis hermanos y, sobre todo, a mi madre. Por su incesante fe en lo que propongo y lo que hago, que ha hecho la experiencia muchísimo más positiva, por ser la víctima de mis eternas propuestas y preocupaciones con el proyecto y por ser la persona más importante de mi vida.

Resumen

El mundo del fútbol genera en sus aficionados una pasión poco comparable con otros medios de entrenimiento en el planeta. Miles de millones de aficionados que ponen las expectativas y la fe en aquellos equipos a los que animan y sobre todo en sus jugadores. Jugadores que, a día de hoy, mueven cifras muy grandes de dinero entre clubes y no dejan de ser una inversión en capital humano con el que generar un impacto positivo para el club.

Sin embargo, esto no siempre es así. Y es que la especulación de jugadores en prensa y aficionados es algo inevitable, estas inversiones millonarias sobre los que se deposita una confianza muy grande y aterrizan en el club bien por ser una promesa de futuro, por ser un refuerzo o por ser una gran estrella se espera que rindan lo mejor posible, y algunas veces acaban siendo mucho menos efectivos en su rendimiento en el club de lo que en su día se comentaba acerca de ellos.

En este proyecto se busca aportar un enfoque analítico donde se ponga en consideración no solo el valor estadístico generado por el jugador, sino además el equipo donde lo hizo, sus características, nivel y estilo de juego. De esta manera se intenta que ojeadores y aficionados puedan reforzar su criterio a la hora de valorar jugadores. Además de esto, se incorporan modelos de aprendizaje automático para ver si la inteligencia artificial es capaz de ver más allá del año de análisis y puede predecir el rendimiento al año siguiente.

Palabras clave

Ciencia de datos, fútbol, rendimiento, web scraping, aprendizaje automático

Abstract

The world of football generates a passion in its fans that is rarely matched by any entertainment media on the planet. Billions of fans place their their hopes and expectations on the teams they support and, above all, on their players. Players who, nowadays, require huge transactions between the clubs, a big expense in salaries and are seen as an investment in human capital with which to generate a positive impact for the club.

However, this is not always the case. The speculation of players from the press and the fans is inevitable, these millionaire investments on which a great confidence is placed on a player to land at the club either as a promise for the future, a tactical reinforcement or as a big star. And it is always expected to perform as well as possible, but sometimes they end up being much less effective in their performance at the club than what was once said about them.

This project seeks to provide an analytical approach that takes into consideration not only the statistical value generated by the player, but also the team where he did it, its characteristics, general level and playstyle. In this way, it is intended that scouts and fans can reinforce their criteria when evaluating players. In addition to this, machine learning models are incorporated to see if artificial intelligence is capable of seeing beyond the year of analysis and can predict performance the following year.

Keywords

Data science, football, performance, web scraping, machine learning

ÍNDICE

Listas

Lista de códigos

Lista de figuras

1

Introducción

El mundo de los deportes es seguido por cientos de millones de personas en todo el planeta. El fútbol es, si acaso, el más destacado de entre todos ellos. Alrededor del llamado “deporte rey” hay todo un público de aficionados que celebra las victorias y sufre las derrotas de sus equipos favoritos, esos aficionados que año tras año permiten a su equipo generar los ingresos disponibles para tomar decisiones en beneficio del club. El club puede invertir dinero en mejorar sus instalaciones de entrenamiento, personal técnico, estadio... En resumen, decisiones relacionadas con aumentar su presencia como entidad deportiva. Si bien estas decisiones pueden aportar una ventaja competitiva, tanto la prensa como los aficionados centran sus focos en una única cosa: sus jugadores.

Jóvenes promesas con la esperanza de que exploten todo su potencial, jugadores asentados que se espera de ellos un impacto casi inmediato... Rumores que copan los titulares de prensa y tienen a todo el mundo en vilo durante los periodos de traspasos. Y no es para menos, pues estos movimientos tienen una repercusión enorme para los clubes, ya sea en forma de más éxito en las competiciones o de conseguir activos que desarrollar y poder vender a un precio más caro en el futuro. Alrededor de estos objetivos los clubes tienen montado un cuerpo de ojeadores que se dedican a encontrar estos fichajes, y sin embargo no siempre aciertan. La pregunta es, ¿hay algo que la ciencia de datos pueda hacer para ayudar a un ojeador a evaluar un fichaje?

La respuesta más evidente es que sí. La tendencia que tienen deportes como el baloncesto o el béisbol a darle más importancia a los datos es algo que permea también al fútbol [1,2]. A día de hoy, se han realizado investigaciones relacionadas con predecir el valor de mercado de futuro de potenciales incorporaciones [3–7] y estudios sobre qué posiciones interesan más para invertir y métodos de análisis de jugadores que permiten a los equipos encontrar mejor qué perfiles interesa fichar [8–11], la gran mayoría con el uso de sets de datos muy exhaustivos de proveedores de pago como pueden ser Opta y Wyscout [12, 13] o mediante parámetros de jugadores para juegos de fútbol como FIFA o Football Manager [3, 6, 14–16].

En resumen las aportaciones de este proyecto son:

- La creación de un **dataset público** con información estadística de jugadores y equipos. Este dataset actualmente se encuentra en la siguiente dirección: `https://github.com/InigoG99/whoscored_dataset`.

- El desarrollo de **modelos de regresión** para intentar predecir el rendimiento futuro de jugadores
- El desarrollo de una **herramienta de recomendación** que permite obtener mayor *insights* sobre el estudio de un jugador como un fichaje.

La memoria de este proyecto está compuesta de las siguientes partes. En el **capítulo 2** se hace un análisis del estado del arte del dominio que ocupa al proyecto y se comentan posibles aportaciones e inconvenientes que presentan estos estudios respecto a la intención de este. En el **capítulo 3** se explica el proceso completo de formación del dataset, desde las fuentes utilizadas y el motivo hasta el tratamiento sobre las mismas, además de profundizar sobre la variable a predecir que se usará en el siguiente capítulo. En el **capítulo 4** se profundiza sobre los modelos que proponemos para el problema en cuestión y se comentan resultados y observaciones sobre los mismos. En el **capítulo 5** se pone un ejemplo práctico, analizando las posibles causas del fichaje de Neymar al Barcelona y cómo el modelo estima su rendimiento al año siguiente, los posibles motivos del por qué y el resultado real. Por último, en el **capítulo 6** están las conclusiones y propuestas de trabajo futuro sobre el proyecto

2

Estado del Arte

Como se ha comentado en la introducción, para este trabajo se ha hecho una investigación previa de qué aportaciones ha podido tener la ciencia de datos al mundo del fútbol y al deporte en general. En este capítulo se comenta en mayor detalle los papers más relevantes que se han podido encontrar.

Para la realización de este trabajo se ha realizado un estudio bibliográfico que se detalla a continuación. Se han realizado 2 consultas sobre la base de datos de SCOPUS. Nótese que el protocolo detallado de la búsqueda bibliográfica se encuentra en el anexo A, donde se incluyen las consultas, motores de búsqueda, resultados y criterios de selección. Una vez realizado este proceso nos encontramos con 300 papers relativos al dominio del machine learning aplicado al mundo deportivo y 111 relativos al uso de análisis de redes sociales al mundo de los deportes. De entre los que nos incumben en este trabajo por adecuarse al tipo de datos con los que vamos a trabajar, el tipo de predicción que querían realizar o un análisis que fuera útil de cara a nuestro proyecto, realizamos la lectura de aproximadamente unos 42 papers. En los siguientes párrafos se comentarán los papers que se han considerado más relevantes en base al deporte sobre el que trabajan, similitud con la propuesta del proyecto, recursos y metodología utilizados y resultados obtenidos. Respecto al análisis de redes sociales no se hará mención en todo el trabajo, puesto que formó parte de la investigación pero se descartó tras la lectura sobre el tema y estudios que se habían realizado.

Estudio del valor de mercado de jugadores

Uno de las aplicaciones más comunes de la ciencia de datos al deporte se refleja en aquellos papers que pretender mejorar las estimaciones del valor de los jugadores. Behravan y Razavi [3] utilizan Optimización por Enjambre de Partículas Automático (APSO) [17] para agrupar los datos obtenidos del videojuego FIFA 20 en distintos clústeres, agrupando a los jugadores en base a sus características. En base a ello, utiliza técnicas para encontrar metaheurísticas y determinar cuales son los atributos más relevantes para cada cluster y entrena un algoritmo SVR (*Support Vector Regression*) [18] para cada uno de los clústeres y predecir el valor de mercado de los jugadores. Yigit et al. [5,6] utilizan un modelo combinado con XGBoost [4] y regresión lineal con regularización Lasso sobre los datos de Football Manager y generan un modelo para estimar el valor de los jugadores con un error cuadrático medio por separado de 0.17 y 0.59 para cada modelo respectivamente. Singh y Lamba [7] van más allá de las

características físicas y técnicas de los jugadores y deciden tener en cuenta su popularidad mediática y su estatus, aumentando el coeficiente de determinación (o R^2) de 0.47 a 0.89 mediante la insercion de estos parámetros, concluyendo con que afectan de forma muy significativa a su valor.

Estudios sobre optimización de plantilla

También se ha investigado sobre maneras de optimizar el rendimiento de la plantilla. Una de las aplicaciones donde puede ser muy útil es en la selección de los jugadores para un partido, como es el caso de Rajesh et al. [8], que busca posiciones óptimas para jugadores en base a sus atributos, además de su nacionalidad y edad mediante técnicas de clustering y Random Forest como el mejor clasificador, con un accuracy de 0.83. Carpita et al. [9] estudiaron la influencia de ciertos atributos en las posibilidades de victoria, concluyendo que reforzar la defensa suele aportar un mayor incremento en las opciones de un equipo de mejorar su desempeño. Heggels et al. [10] realizaron un análisis acerca de las posiciones del campo y sus probabilidades de marcar goles y utilizaron algoritmos de predicción para ver si se podía predecir el resultado de los partidos solo en base a las ocasiones generadas y la posición de las mismas con Random Forest [19] obteniendo un área bajo la curva (o AUC) de 0.81. Decroos y Davis [11] utilizaron información espacio-temporal de los partidos con la intención de ser capaces de describir detalladamente el estilo de los jugadores mediante vectores agrupados por zona del campo y tipo de acción que realizan.

Estudios sobre rendimiento de jugadores

Respecto al rendimiento de jugadores, que es el tema que se busca abordar en este trabajo, se han realizado estudios que realizan una evaluación del rendimiento en base distintos parámetros. Decroos et al. [20] evalúan el rendimiento de los jugadores en base a cómo sus acciones se traducen a goles, ya sea anotados o concedidos, en los siguientes movimientos de un partido, utilizando CatBoost para dar ciertos valores de probabilidad a las acciones en la fase de entrenamiento y obteniendo un AUC de 0.77 para las siguientes acciones conducen a gol anotado y 0.73 para acciones que conducen a gol concedido. Gracias a ello, pueden encontrar jugadores valiosos en base a su aportación en goles y/o asistencias, jugadores prometedores debido a su aportación a goles y asistencias y su edad y posibles reemplazos de otros jugadores en base al valor que generan por tipo de acción cada 90 minutos.

Del mismo modo, Bransen y Van Haaren [21] utilizan XGBoost para estudiar las contribuciones de los pases de un jugador a goles para sus equipos y ofrecer reemplazos de jugadores en base a cómo de similares son sus contribuciones. Kim et al. [22] proponen un modelo que permite evaluar el rendimiento de los fichajes en base a su valor en el mercado y su rendimiento e intentar predecir como puede funcionar un jugador. Tomando los datos del proveedor Wyscout [13], realizan una regresión lineal para intentar predecir cuales serian sus números de cara a la temporada siguiente, utilizan Boruta [23] para seleccionar los atributos más importantes por posición y en base a ellos utilizar K-Means [24] para clasificar en función de su coste de traspaso y de su rendimiento predicho. Las limitaciones

encontradas se comentan en la siguiente sección.

Por último, Pappalardo et. al [25] utilizan un dataset de eventos espacio-temporales en partidos a lo largo de una temporada y en base a las acciones ocurridas y el resultado del partido en victoria o derrota realizan una medición del rendimiento de un jugador en función de un sistema propio que se basa en el peso de sus acciones de cara a ayudar a sus equipos a conseguir victorias. Los pesos de las acciones se diferencian en base a la posición y se han determinado mediante la generación de un modelo de clasificación lineal mediante vectores de soporte (*LSVC*) [26] por posición, obteniendo como raiz normalizada del error cuadrático medio (*NRMSE*) un 15 %. Sobre los pesos obtenidos por posición se ha realizado una valoración de los jugadores. Este último trabajo es especialmente potente, pero carecemos de los recursos necesarios para obtener más de una temporada con este sistema, y el proyecto está basado en predecir el rendimiento futuro, por lo que a falta de fuentes de pago más completas, se descartó hacer una aportación sobre este trabajo.

2.1. Oportunidad Científica

Dicho esto, hay una cuestión en la que no se ha incidido de cara a la evaluación de fichajes, y es que no se tiene en especial consideración el estilo de juego del equipo en el que se encuentra de cara a su rendimiento. Esta cuestión para nosotros es importante, puesto que consideramos que el contexto táctico donde ese jugador va a poner sus habilidades al servicio del club repercute de forma significativa en su rendimiento. Pongamos el ejemplo de un jugador que rinde de forma excepcional en equipos verticales y de contraataque por su velocidad. Claramente, este arquetipo de jugador verá mermada su aportación en un contexto de equipos más centrados en la retención de posesión y con un ritmo más pausado, puesto que esa característica que tanto aportaba al primer esquema tiene una relevancia algo menor en el segundo. Por ello, se abre la posibilidad de estudiar el rendimiento de los jugadores en base a los datos de su equipo además de sus estadísticas personales.

Otro valor importante que no se menciona es el nivel del equipo a nivel global, puesto que todo el análisis de rendimiento que se ha hecho es en un vacío que no pone el contexto la exigencia media de la competición. Esto, si bien es muy difícilmente medible con precisión debido a que no se pueden sacar conclusiones en base a hechos empíricos sino que se se basa en las hipótesis(es decir, estas valoraciones no salen de un contexto de competición entre todos los equipos a nivel mundial), aportar una métrica respecto al nivel del equipo y escalar el rendimiento en base a ello será objeto de estudio de este trabajo, puesto que se cree que tiene más dificultad rendir mejor a más exigente es el nivel medio del equipo en el que juega un jugador.

Con todos estos datos podremos estudiar mejor qué fichajes han funcionado bien según el contexto de juego actual, el nivel del equipo en el que están y en el que van a estar la siguiente temporada, y de esa forma no solamente basarse en similitud con otros jugadores o aportación a goles [20], sino en

cómo han podido funcionar otras situaciones similares a ellos, ya sea cambiando de equipo o no.

Otras cuestión que, sin ser un problema, consideramos muy relevante mencionar está relacionado con las fuentes de datos que se han utilizado en las investigaciones mencionadas. Algunas de ellas se sustentan en atributos físicos, mentales y técnicos que son otorgados a los jugadores por profesionales dedicados a valorar las habilidades de los mismos para videojuegos. Ejemplos de ellos son los datos de los videojuegos FIFA y Football Manager [27, 28], que han sido usados de forma recurrente debido a que ofrecen una valoración estandarizada de los jugadores y permiten acceder a variables que no suelen aparecer en una tabla estadística como pueden ser su velocidad, capacidad de definición, etc. Sin poner en duda las habilidades de los profesionales encargados de esta labor, es cierto que muchas de estas valoraciones muchas veces son criticadas por la comunidad y están supeditadas a las particularidades del motor gráfico del juego. La saga de videojuegos FIFA ha sido recientemente criticada por parte de sus jugadores [29] por imprecisión. Además de ello, consideramos que estos parámetros no responden de forma directa al rendimiento del jugador durante una temporada, habiéndose equivocado, por ejemplo, en predecir máximos goleadores en base a sus atributos ofensivos con mucha frecuencia [30]. Por tanto, tampoco responden a cómo ha rendido un jugador durante una temporada.

Otras fuentes de datos que hemos mencionado sí que responden a nuestras necesidades con creces, que son las de proveedores privados como Opta o Wyscout entre otros. Estas fuentes ofrecen información exhaustiva de los jugadores que cubren y sin duda hubieran aumentado la calidad del trabajo de forma significativa. Sin embargo, no conseguimos que nos facilitaran muestras de datos con fines académicos, y consideramos tremendamente excesiva la inversión que suponía el acceso, siendo un trabajo realizado por un estudiante. Agradecer al Dr. Luca Pappalardo, de la Universidad de Pisa, por facilitarnos el acceso a un dataset con eventos espacio-temporales [31], que pese a ser un dataset de una calidad excelente, la falta de temporadas adicionales en los datos nos obligó a buscar alternativas gratuitas y de menor información.

Hipótesis de trabajo

En este trabajo se busca desarrollar un sistema de predicción donde, dado el histórico de un jugador hasta el momento y los datos del equipo donde se encuentra, se predice el rendimiento que tendría y se compara con la predicción de rendimiento que resultaría de solo entrenarse con los datos estadísticos del jugador. Con esta aportación, se desarrolla un sistema de soporte a las decisiones (DSS) donde ojeadores y analistas de clubes deportivos podrían evaluar con el mayor criterio posible el rendimiento de numerosos jugadores en su equipo de cara al próximo año y orientar sus fichajes en base a ello.

3

Dataset

En esta sección se procederá a explicar con detalle el proceso que se ha seguido para obtener el dataset resultante. Como se mencionó anteriormente, este dataset está compuesto por un histórico de temporadas hasta la temporada 2009/2010 de los jugadores que están actualmente compitiendo en alguna de las ligas más importantes del planeta.

En resumen, este dataset está compuesto de 3 fuentes diferentes:

Estadísticas de jugador por temporada y **Rating**(ver sección 3.5)

Puntuación media de equipos por temporada

Estadísticas de equipo por temporada

Cada fila del dataset va a estar compuesto por un vector $x = (x_{jug}, x_{punt}, x_{eq})$ y una variable y.

- El subvector x_{jug} hace referencia a los datos de un jugador por temporada y su rendimiento en forma de Rating. Está compuesto de x columnas y su extracción y tratamiento se detalla en la sección 3.1.

- El subvector x_{punt} hace referencia a la fortaleza de los equipos en forma de puntuación global de los equipos a lo largo de una temporada. Está compuesta solo 1 columna y todo el proceso relacionado con este dato se explica en la sección 3.2.

- El subvector x_{eq} hace referencia a los datos de equipo por temporada. Compuesto por x columnas, se describe con profundidad en la seccion 3.3.

- La variable y será el Rating de un jugador la temporada siguiente, constituirá nuestra variable a predecir y se habla de ella con detalle en la sección 3.5.

En primer lugar se considera conveniente explicar el por qué de las ligas que hemos elegido. Para empezar, este trabajo pone el foco en las ligas europeas, puesto que son con diferencia las más populares y las que más dinero mueven, y por ello donde la gran parte del talento mundial está concentrado. Este dataset está compuesto por los equipos de las 10 ligas más importantes del continente, de acuerdo con el ránking establecido por UEFA [32]. Estos países son: Alemania, Austria, Bélgica, España, Francia, Holanda, Inglaterra, Italia, Portugal y Rusia.

Además de estos países, se ha decidido incorporar a las primeras divisiones de Brasil y Argentina al dataset, puesto que son los dos países de América que más jugadores exportan a estas ligas [33] y los dos países no europeos mejor situados en el ranking FIFA [34]. También se consideró incluir Colombia y Uruguay por los mismos motivos, pero las fuentes de datos usadas ofrecían muy poca información sobre estas ligas, por lo que quedaron descartadas.

Todo este trabajo de fusión, tratamiento y análisis de datos ha sido realizado sobre la librería de *python3* **pandas**, que es una librería enfocada al procesamiento de volúmenes grandes de datos y nos permite manipular de forma muy efectiva todos los datos que manejamos en este proyecto.

En los siguientes apartados del tema se abordará en profundidad cada una de las partes.

3.1. Estadísticas de jugadores (x_{jug})

Este es el elemento central alrededor del que orbita todo el dataset. En esta sección se toman las estadísticas por temporada de los jugadores que se encontraran jugando en las ligas previamente mencionadas. Las temporadas abarcan desde el año 2019/2020 hasta, como máximo, la temporada 2009/2010.

La fuente que se ha decidido tomar para estos datos es la página WhoScored [35], que contiene una cantidad de datos muy grande y detallada de jugadores y temporadas de forma completamente gratuita, además de una variable propia de la página que será fundamental en los siguientes puntos del trabajo que es el *Rating*, sobre el que se hace un análisis en la sección 3.5.

La extracción relacionada con esta fuente está inspirada en un repositorio encontrado en la página GitHub, realizado por José Ramón Arias [36]. Dicho proyecto ha servido como base para crear un sistema de webscraping de esta página que cumpliese con las exigencias de este trabajo. El estado inicial del proyecto mencionado estaba poco acorde con lo que se buscaba, puesto que estaba desactualizado con la estructura actual de la fuente, por lo que ha tenido que sufrir por bastantes arreglos. Además, se ha incorporado sobre este proyecto funcionalidad adicional que permitiera recopilar más datos de jugadores de lo que el proyecto inicial podía.

3.1.1. Extracción

Desarrollo del *web scraping*

Lo primero que se hizo fue realizar una primera extracción de los enlaces de equipos que conforman las ligas que se van a usar mediante un script que recorrerá las secciones de la web con las tablas de las ligas que se quiere utilizar. Una vez hecho eso y obtenidas las URLs de todos los equipos que nos interesaban, se obtienen las URLs de los jugadores de todos los equipos accediendo a la sección de

cada equipo y buscando en sus plantillas.

Una vez hecho eso, se desarrolló un script que recorre todas las URLs obtenidas. La estructura de la tabla estadística está compuesta de distintos botones que alteran el contenido de la tabla y muestran información de distintas secciones. En el script se incorpora funcionalidad que simula la interacción con esos botones mediante Selenium [37], puesto que el HTML de la página es dinámico y no se pueden obtener todos los datos solo con una carga del contenido de la página.

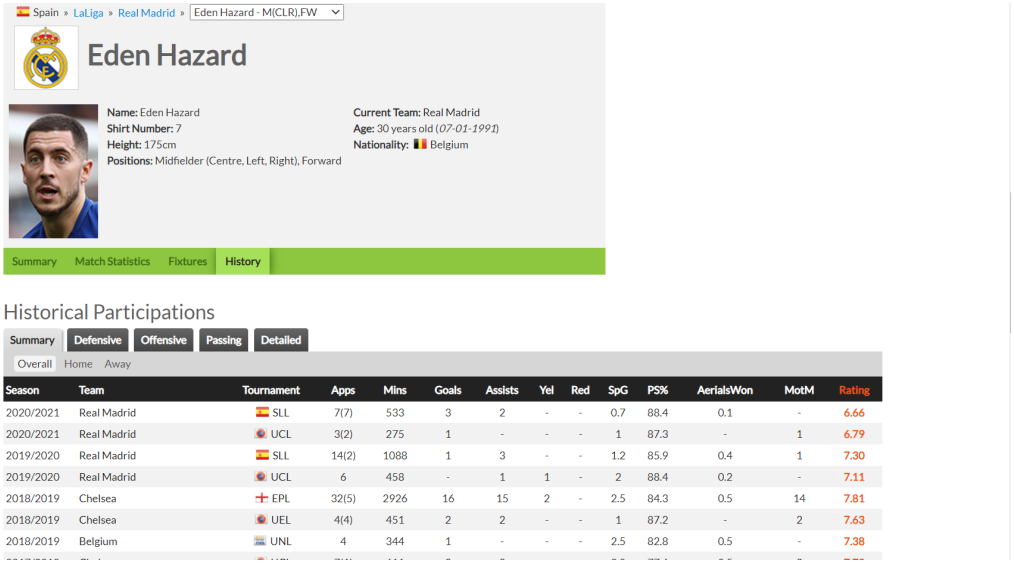


Figura 3.1: Ejemplo de página con el histórico de un jugador. Fuente: *whoscored.com*

El vector está compuesto de 28 columnas, estructurado de la siguiente manera, siendo cada subvector una de las secciones disponibles que tiene la página, disponible:

$$x_{jug} = (x_{sum}, x_{def}, x_{off}, x_{pass}, x_{det})$$

Estas secciones son:

Summary (x_{sum}): Resumen estadístico. Datos extraídos: Equipo, Competición, Temporada, Goles, Asistencias, Tarjetas amarillas, Tarjetas rojas, Disparos, % de pases completados, **RATING**. Total: 9 columnas,

Defensive (x_{def}): Estadísticas relacionadas con el rendimiento defensivo. Datos extraídos: Entradas, Intercepciones, Faltas realizadas, Fueras de juego provocados, Despejes, Nº de veces regateado, Bloqueos. Total: 7 columnas.

Offensive (x_{off}): Estadísticas relacionadas con el rendimiento ofensivo. Datos extraídos: Regates realizados, Faltas provocadas, Nº de fueras de juego cometidos, Nº de posesiones perdidas. Total: 4 columnas.

Passing (x_{pass}): Estadísticas relacionadas con los pases ejecutados. Datos extraídos: Pa-

ses clave, Pases realizados, Centros, Pases largos, Pases infiltrados. Total: 5 columnas

Detailed (x_{det}): Desglose de los tiros realizados por zona del campo. Datos extraídos: Tiros fuera del área, Tiros en el área pequeña, Tiros en área de penalti. Total: 3 columnas.

Nótese que en el anexo B figura una descripción detallada de todos y cada uno de los datos que contiene.

3.1.2. Tratamiento

En esta sección se comenta de forma extensa los cambios que se han realizado sobre algunas columnas y filas una vez se han tomado los datos de cara a conformar una fuente lo más estable posible y que permita su fusión con las otras.

Agregado de competiciones por equipo

En primer lugar, en el apartado anterior se menciona que cada fila contiene datos por temporada, equipo y competición. Este último punto es el que más problemas causa en primera instancia, puesto que interesa tener a cada jugador con un valor único por temporada y equipo. Si bien es cierto que hay que aclarar que no todas las competiciones son iguales (ej. las competiciones continentales suelen ser de mayor exigencia que las ligas locales debido a que en las primeras juegan los mejores de cada liga local), se consideró que la presencia de múltiples filas por jugador, temporada y equipo dificulta mucho el análisis. Como solución a esto, se ha decidido agrupar los datos por temporada, jugador y equipo y sumar los datos en caso de que sean números naturales y en caso de porcentajes y puntuaciones se hace una suma ponderada por minutos jugados en cada competición.

Aparte está el hecho de que el jugador pueda ser internacional y figure su selección como equipo en nuestro dataset, cuyas filas han sido eliminadas comprobando si su nacionalidad y su equipo coinciden o si su equipo contiene las palabras U21 y U19, que se refieren a categorías inferiores.

Eliminación de claves primarias duplicadas

El proceso anterior solamente ha permitido tener como clave primaria (campos que permitan identificar una fila de forma única) el nombre del jugador, la temporada y el equipo en el que estaba. Esto, en una situación en la que pudiésemos saber con certeza la linea temporal del jugador, sería idóneo. Primero, porque se dispondrían más filas en las que trabajar, y segundo porque así figurarían en nuestro dataset movimientos ocurridos ya sea al final del mercado de verano o en el mercado de invierno. Sin embargo, se concluyó con que, de cara a predecir el rendimiento futuro, tener varias filas con la misma temporada para un jugador podría ser conflictivo. Finalmente, se decidió tomar solo la parte de la temporada en el equipo donde ese jugador haya jugado más minutos.

Estandarización de temporadas

Un pequeño detalle que se encontró es que las temporadas de los equipos de la liga brasileña están indicadas de forma diferente. Mientras que los demás equipos tienen por temporada el formato *<año de comienzo/año de finalización>*, la fuente tiene tomadas las brasileñas solo por el año de comienzo, así que por estandarización y de cara a realizar operaciones que requieren de esta información, se decidió cambiar el formato para que se adecuara con el resto.

Mapeado de posiciones

Otro problema que se ha encontrado está relacionado con las posiciones, puesto que figuran en la ficha de cada jugador como una cadena de texto, en ella figuran las posiciones más habituales del jugador, separadas por secciones del campo mediante comas (en líneas generales, aunque no siempre se cumple). Un jugador puede cubrir más de una posición por sección, indicadas entre paréntesis, y más de una sección. De cara a poder interpretar esa cadena de texto de alguna forma para que pudiera ser interpretada por algoritmos de aprendizaje automático de forma más estandarizada, el sistema de meapeado utilizado se ha basado en el concepto de **One-Hot Encoding**, que consiste en generar una variable binaria por cada opción posible que haya dentro de una variable categórica. La diferencia de este método respecto al que se usa aquí es que normalmente de esa cadena de texto de posiciones es muy común que salgan más de una variable binaria codificada a 1, en contraposición a lo habitual que es más una relación 1 a 1.

Las posiciones que se han considerado son:

GK: Portero(*Goalkeeper*)

CB: Defensa central(*Centre-back*)

LB: Lateral izquierdo(*Left-back*)

RB: Lateral derecho(*Right-back*)

CDM: Mediocentro defensivo(*Centre Defensive Midfielder*)

CM: Mediocentro(*Midfielder*)

CAM: Mediocentro ofensivo(*Centre Attacking Midfielder*)

LW: Extremo izquierdo(*Left Winger*)

RW: Extremo izquierdo(*Right Winger*)

FW: Delantero centro(*Forward*)

Esto añade a la fuente 10 columnas más, haciendo un total de 38 columnas las que contiene esta fuente.

Eliminación de nulos

Debido a la falta de información en algunos datos de temporadas de jugador, se ha decidido eliminar esas columnas con datos nulos del dataset, puesto que en lineas generales faltan datos por estar en equipos externos a estas ligas, en las que la fuente no recoge toda la información que se usa.

3.2. Puntuación de equipo (x_{punt})

Una de las cosas que se consideró importante es reflejar de alguna manera el nivel de competición en el que un jugador está desarrollando sus habilidades, puesto que un rendimiento notable no tiene el mismo valor en un equipo de una liga muy exigente y que disputa las mayores competiciones del planeta que en un equipo cuyo techo competitivo es limitado. Es por ello que entra la segunda fuente de datos al dataset, la puntuación de equipo. La idea principal de esta fuente es asignar un valor numérico a la fortaleza que tienen los distintos equipos de nuestro dataset para así tener una idea un poco más informada del nivel del equipo donde los jugadores producen sus números a lo largo de la temporada.

Esta información de base no está respaldada por hechos claros, puesto que pretende colocar en un mismo ránking a equipos que rara o nula vez tienen la oportunidad de encontrarse. ¿Cómo se medir en realidad cuan bueno es un equipo como el Boca Juniors (Argentina) a nivel mundial? Es un equipo que compite todas las temporadas por los primeros puestos en una liga y que a excepción de, como mucho, un partido al año, no tiene la oportunidad de enfrentarse a equipos europeos. A menos que se pusiera a este equipo a competir en todas las ligas, nunca se sabría con certeza qué tan bueno es respecto del resto. En competiciones continentales se puede obtener algo más de información acerca de cómo están ambos equipos, pero al no estar en un formato de liga regular, sino en torneos por eliminación, toda la información se basa en su éxito en esas competiciones e hipótesis.

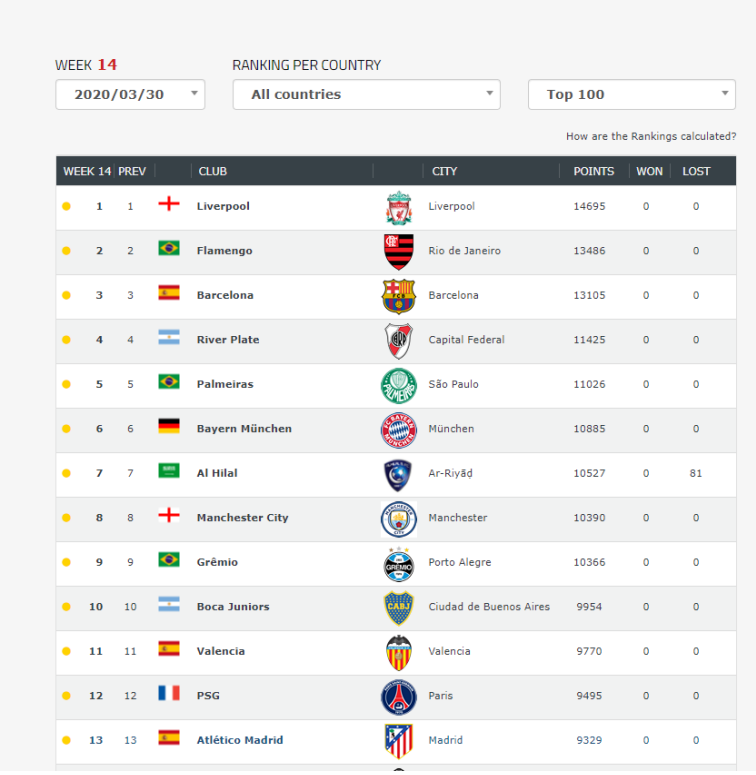
Sin embargo, la fuente de la que se extrae este dato hace, en nuestra opinión, un muy buen trabajo. La fuente en concreto proviene del sitio *clubworldranking.com* [38], que contiene un ranking de los 1000 mejores equipos por semanas, utilizando un sistema de puntos que procederemos a explicar de forma breve para aportar justificación a la elección de esta fuente.

El sistema otorga a un equipo puntos tras un partido en base a los siguientes parámetros

Resultado del partido: 3 puntos por ganar, 1 punto por empatar, 0 por perder

Importancia del partido: Multiplicador en base al tipo de competición y ronda del partido. Ver Anexo para descripción detallada.

Fortaleza de competición: Multiplicador que tiene en cuenta el coeficiente de confederaciones de un país, que se calcula en base al rendimiento de sus equipos en competiciones internacionales en los últimos 5 años. (solo partidos domésticos)



WEEK 14 | RANKING PER COUNTRY

2020/03/30 | All countries | Top 100

How are the Rankings calculated?

WEEK 14	PREV	CLUB	CITY	POINTS	WON	LOST
1	1	Liverpool	Liverpool	14695	0	0
2	2	Flamengo	Rio de Janeiro	13486	0	0
3	3	Barcelona	Barcelona	13105	0	0
4	4	River Plate	Capital Federal	11425	0	0
5	5	Palmeiras	São Paulo	11026	0	0
6	6	Bayern München	München	10885	0	0
7	7	Al Hilal	Ar-Riyāḑ	10527	0	81
8	8	Manchester City	Manchester	10390	0	0
9	9	Grêmio	Porto Alegre	10366	0	0
10	10	Boca Juniors	Ciudad de Buenos Aires	9954	0	0
11	11	Valencia	Valencia	9770	0	0
12	12	PSG	Paris	9495	0	0
13	13	Atlético Madrid	Madrid	9329	0	0

Figura 3.2: Imagen con las puntuaciones de equipos en la semana del 30-03-2020 Fuente: *club-worldranking.com*

Equilibrador de partidos: Parámetro de ajuste que tiene en cuenta los partidos jugados por competición doméstica y los sincroniza a 38 partidos. (solo partidos domésticos)

Fortaleza oponente: Multiplicador que tiene en cuenta el coeficiente de país del oponente al que se enfrenta (solo partidos internacionales)

Fortaleza confederación FIFA: Multiplicador que tiene en cuenta la confederación FIFA donde se está dando el partido.

Gracias a esta fuente figuran las puntuaciones de los 1000 mejores equipos entre la semana del 5 de diciembre de 2011 hasta la semana del 30 de marzo de 2020. La agrupación que se ha realizado es por temporada, para lo que se ha asumido que una temporada está compuesta de la segunda mitad de un año y la primera mitad de la siguiente, se toman las puntuaciones de cada equipo durante ese periodo y se hace la media de esas puntuaciones.

3.3. Estadísticas de equipo (x_{eq})

La última fuente de datos permite poner en clave la situación en la que el jugador ha rendido de la forma que lo ha hecho durante una temporada. Obviamente, esto hace referencia a las estadísticas de equipo, que procederemos a explicar qué datos contienen, no sin antes comentar aspectos

relacionados con el proceso de extracción y algunas características importantes.

La fuente de donde han sido obtenidos estos datos es, de nuevo, la web de WhoScored.com. De forma similar a los jugadores, es una fuente que recopila un número grande de estadísticas, desde datos númericos como goles y disparos por partido hasta porcentajes de acciones por tipo, que permiten dar una idea de cómo juega un equipo.

Temporadas disponibles por país

Este nivel de detalle en las estadísticas, sin embargo, conlleva a un inconveniente muy importante, y es que esta recolección estadística es muy inconsistente en cada liga. Debido a ello, hay ligas como las conocidas “5 grandes” (Alemania, Francia, España, Inglaterra e Italia) que están documentadas en todas las temporadas de jugadores disponibles, mientras que la cantidad de temporadas disponibles de otros equipos es muy variable.

Están presentes, por cada país, datos completos a partir de la siguientes temporadas:

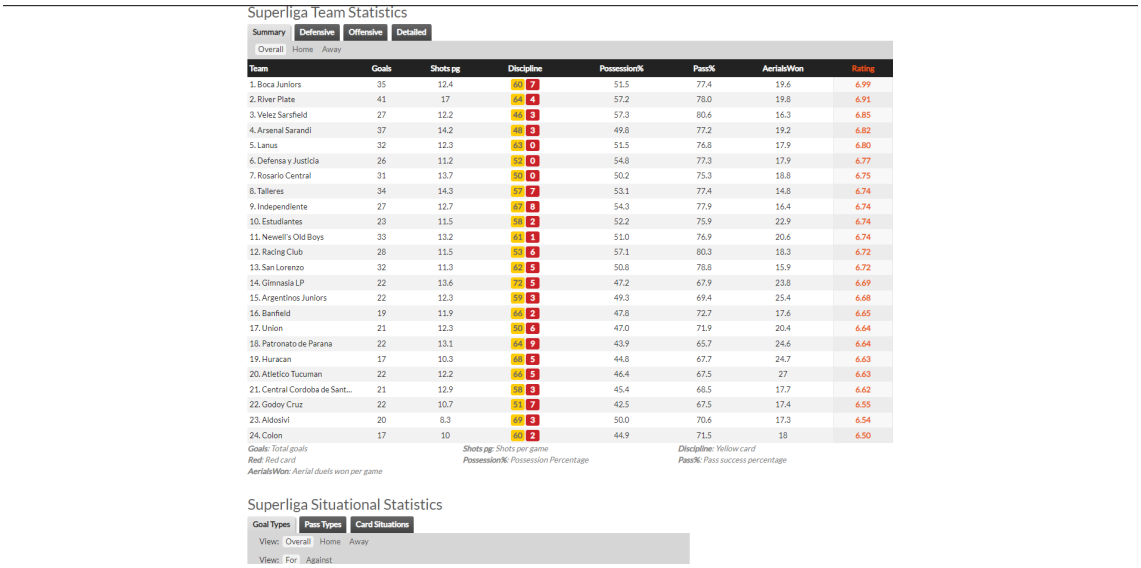
- Alemania: 2010/2011
- Argentina: 2016/2017
- Austria: No tiene
- Bélgica: 2020/2021
- Brasil: 2013/2014
- España: 2010/2011
- Francia: 2010/2011
- Holanda: 2013/2014
- Inglaterra: 2010/2011
- Italia: 2010/2011
- Portugal: 2016/2017
- Rusia: 2013/2014

Sin embargo, se consideró que, pese a que tener datos incompletos de esas temporadas, las pertenencientes a las ligas más importantes cubren todo el rango de temporadas que permitían el resto de fuentes, por lo que se ha decidido que seguía constituyendo un volumen dentro de lo aceptable.

3.3.1. Extracción

De forma similar a como se hizo con los jugadores, utilizamos técnicas de webscraping para obtener las URLs de las temporadas disponibles de cada liga. Una vez obtenidas, el script recorre la pestaña "Team Statistics"de todas ellas, en la que figuran los siguientes campos.

Estadísticas de equipo



Superliga Team Statistics

Summary | Defensive | Offensive | Detailed

Overall | Home | Away

Team	Goals	Shots pg	Discipline	Possession%	Pass%	AerialsWon	Rating
1. Boca Juniors	35	12.4	60 7	51.5	77.4	19.6	6.99
2. River Plate	41	17	64 4	57.2	78.0	19.8	6.91
3. Velez Sarsfield	27	12.2	46 3	57.3	80.6	16.3	6.85
4. Arsenal Sarandi	37	14.2	48 3	49.8	77.2	19.2	6.82
5. Lanus	32	12.3	43 0	51.5	76.8	17.9	6.80
6. Defensa y Justicia	26	11.2	52 0	54.8	77.3	17.9	6.77
7. Rosario Central	31	13.7	50 0	50.2	75.3	18.8	6.75
8. Talleres	34	14.3	57 7	53.1	77.4	14.8	6.74
9. Independiente	27	12.7	67 8	54.3	77.9	16.4	6.74
10. Estudiantes	23	11.5	58 2	52.2	75.9	22.9	6.74
11. Newell's Old Boys	33	13.2	41 1	51.0	76.9	20.6	6.74
12. Racing Club	28	11.5	53 6	57.1	80.3	18.3	6.72
13. San Lorenzo	32	11.3	42 5	50.8	78.8	15.9	6.72
14. Gimnasia LP	22	13.6	72 5	47.2	67.9	23.8	6.69
15. Argentinos Juniors	22	12.3	59 3	49.3	69.4	25.4	6.68
16. Banfield	19	11.9	66 2	47.8	72.7	17.6	6.65
17. Union	21	12.3	50 6	47.0	71.9	20.4	6.64
18. Patronato de Parana	22	13.1	64 9	43.9	65.7	24.6	6.64
19. Huracan	17	10.3	68 5	44.8	67.7	24.7	6.63
20. Atletico Tucuman	22	12.2	66 5	46.4	67.5	27	6.63
21. Central Cordoba de Sant...	21	12.9	58 3	45.4	68.5	17.7	6.62
22. Godoy Cruz	22	10.7	51 7	42.5	67.5	17.4	6.55
23. Aldosivi	20	8.3	69 3	50.0	70.6	17.3	6.54
24. Colon	17	10	60 2	44.9	71.5	18	6.50

Goals: Total goals
Red: Red card
AerialsWon: Aerial duels won per game
Shots pg: Shots per game
Possession%: Possession Percentage
Discipline: Yellow card
Pass%: Pass success percentage

Superliga Situational Statistics

Goal Types | Pass Types | Card Situations

View: Overall | Home | Away

View: For | Against

Figura 3.3: Ejemplo reducido de una página con las estadísticas de la liga. Fuente: *whoscored.com*

Summary: Resumen estadístico por temporada, equipo y competición. Datos extraídos: Nombre de equipo, Posición en liga, Goles, Disparos por partido, Tarjetas amarillas, Tarjetas Rojas, % de posesión, % de pases con éxito, Duelos aéreos ganados, Rating promedio del equipo. Total: 8 columnas.

Defensive: Estadísticas relacionadas con el rendimiento defensivo. Datos extraídos: Tiros permitidos por partido, Entradas por partido, Intercepciones por partido, Faltas por partido, Fueras de juego provocados por partido. Total: 5 columnas.

Offensive: Estadísticas relacionadas con el rendimiento ofensivo. Datos extraídos: Disparos a puerta por partido, Regates por partido, Faltas recibidas por partido. Total: 3 columnas.

Detailed: Desglose de los tiros realizados por zona del campo. Datos extraídos: Tiros fuera del área, Tiros en el área pequeña, Tiros en área de penalti. Total: 3 columnas.

Estadísticas situacionales

Goal Types: Goles anotados y concedidos desglosados por la situación de partido en la que se da. Datos extraídos: Goles en jugada, Goles al contraataque, Goles a balon parado, Penaltis, Goles en propia. Total: 5 x 2 = 10 columnas.

Pass Types: Media de pases realizados y recibidos por partido agrupados por tipo de pase. Datos extraídos: Centros por partido, Pases infiltrados por partido, Pases largos por partido, Pases cortos por partidos. Total: 4 x 2 = columnas.

Estadísticas posicionales

Attack Zones: Porcentaje de ataques realizado por lado del campo. Datos extraídos: Ataques por la izquierda, Ataques por el medio, Ataques por la derecha. Total: 3 columnas.

Shot Directions: Porcentaje de tiros realizados y recibidos por lado del campo. Datos extraídos: Tiros por la izquierda, Tiros por el medio, Tiros por la derecha. Total: 3 x 2 columnas.

Action Zones: Porcentaje de acciones realizadas según el tercio de campo donde se produzca. Datos extraídos: Acciones en tercio defensivo, Acciones en tercio medio, Acciones en tercio ofensivo. Total: 3 columnas

Una descripción del significado de todas estas columnas se detalla en el anexo B.

3.4. Tratamiento adicional y resumen

Una vez comentadas todas la fuentes de datos, se tiene que juntar de forma compacta, de forma que cada fila contenga las estadísticas de un jugador en una temporada y su Rating, la puntuación de equipo en el que estuvo, las estadísticas del equipo y (a poder ser) el equipo, puntuación y Rating de la temporada siguiente.

Estandarización de nombres de equipo

Esto en primera instancia suena sencillo, sin embargo, las 3 fuentes presentan una inconsistencia que no permite sacar partido al máximo de las mismas. Esta es el nombre del equipo.

En cada una de las distintas fuentes se encuentra que equipos iguales tienen distintos nombres según la fuente. Un ejemplo de ellos sucede con el Manchester City, que mientras que en dos fuentes figura con ese nombre, en otra aparece con el nombre “Man City”. Esto en los equipos más conocidos es una tarea que no requiere de mucho tiempo corregirla a mano. Sin embargo, la cantidad tan grande de equipos que manejan todas las fuentes es muy grande como ir caso por caso.

Es por ello que se ha hecho uso de la librería **difflib**. Esta librería incorpora funciones que permiten hacer comparaciones de secuencias y encontrar diferencias en ellas. Mediante el uso de esta, se puede analizar todas las cadenas de texto correspondientes a los equipos de todas las fuentes y utilizar la función de **get_close_matches**, que saca las cadenas más similares entre unas y otras. Se ha generado de esta manera en un tiempo razonable un diccionario con la mayor parte traducciones directas de los nombres de los equipos de la fuente de las puntuaciones y de las estadísticas de equipo a los nombres que figuran en la fuente de datos de los jugadores.

Resultado final del dataset

Realizado todo el dataset, se tienen los siguientes datos:

17589 filas

97 columnas

6940 nulos en Next Rating, 2831 nulos en todas las estadísticas de equipo (x_eq), 5396 en Next Team

Player Name	Season	Team	Rating
Lionel Messi	2011/2012	Barcelona	106.150508
Lionel Messi	2019/2020	Barcelona	103.631592
Lionel Messi	2014/2015	Barcelona	103.331773
Robert Lewandowski	2019/2020	Bayern	103.248752
Neymar	2017/2018	PSG	102.496264
Lionel Messi	2018/2019	Barcelona	100.543864
Cristiano Ronaldo	2013/2014	Real Madrid	99.949068
Lionel Messi	2012/2013	Barcelona	99.109486
Lionel Messi	2017/2018	Barcelona	98.620770
Lionel Messi	2015/2016	Barcelona	96.765988

Figura 3.4: Top 10 Rating del dataset

3.5. Variable a predecir: Next Rating

Explicación del Rating

Anteriormente, más concretamente en el punto 3.1.1, se comentaba qué datos se habían sacado de los jugadores para el dataset, en ellos se menciona uno que no tiene una traducción directa a terminología de fútbol. Esa variable es el Rating.

El Rating [39] determina con una nota de entre 0 y 10 el rendimiento de un jugador en un partido. Este rating es calculado por la propia WhoScored en base a los datos provistos por Opta y es considerado uno de los mejores indicadores de rendimiento que hay, utilizado por prensa y clubes entre otros.

El valor parte al inicio de cada partido desde un 6.0, y se tienen en cuenta más de 200 estadísticas a la hora de dar con la estimación, tanto de esta variable como la de la del propio Rating del equipo.

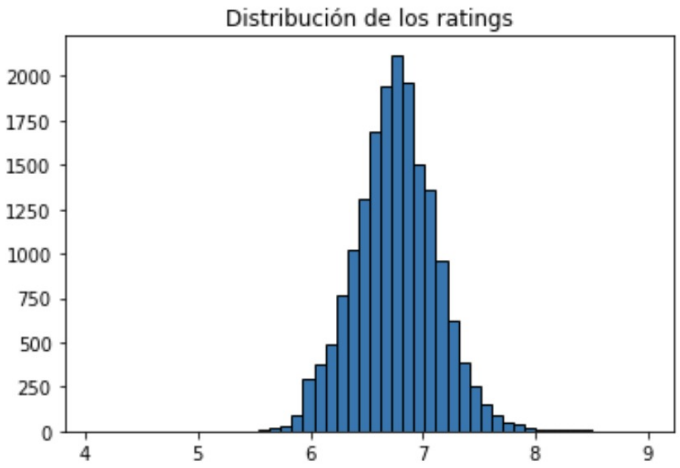


Figura 3.5: Distribución de la variable Rating previa al escalado.

Cada acción del partido puede tener un efecto tanto positivo como negativo y se evalúa cada una en función de la zona del campo donde esa acción haya acontecido y de su repercusión.

Es por ello que se consideró que esta variable nos ofrece un reflejo bastante preciso de como está rindiendo un jugador, cuyo Rating en nuestro dataset refleja el Rating promedio de ese jugador en todos los partidos que ha disputado.

Next Rating

La idea central del proyecto reside alrededor de estimar como de bueno podría ser un fichaje, por lo que es natural que una de las principales propuestas del mismo sea desarrollar un modelo predictivo que, dadas las estadísticas de un jugador durante un año, se pueda predecir con una precisión razonable su rendimiento en el siguiente año.

Lo primero de todo es que, como se puede ver en la figura 3.5, hay una concentración de valores en torno al intervalo [6.5, 7.0] en la distribución. Eso constituye por si solo un problema, puesto que no todos los equipos tienen el mismo nivel y esto solamente evalúa acciones de partido en el vacío. Por lo que se consideró importante hacer un escalado de ese valor en función de otros que consideramos importantes, que son:

- Nivel del equipo en el que juega

- Cuánto destaca ese jugador con respecto a su equipo

Para eso, se escalaron los ratings del año en base a la siguiente fórmula.

$$Rating_{nuevo} = Rating * ln(Puntuacion\_equipo) * \frac{Rating}{Rating_{equipo}}$$

De esta forma se pretende no solo poner en contexto como los jugadores estén rindiendo en base al nivel medio del equipo, sino que se busca hacer resaltar jugadores que estén destacando del resto de su equipo, para ver así con más frecuencia posibles jugadores estrella que no estén en los equipos grandes y se vean mermados por la puntuación de este.

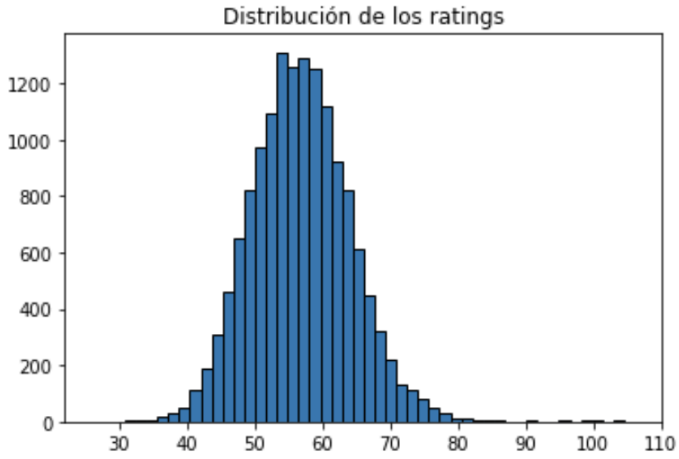


Figura 3.6: Distribución de la variable Rating después del escalado.

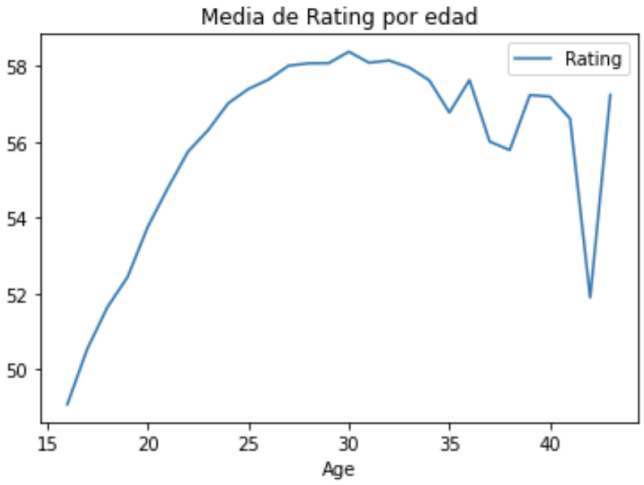


Figura 3.7: Valor medio de la Rating por edad

En la figura 3.7 se puede ver que ciertos atributos como la edad, tienen una relación directa de cara a cómo crece el Rating de un jugador. Esto tiene sentido, puesto que se espera que el rendimiento de un jugador aumente a medida que este va alcanzando su madurez futbolística, además de incrementar un función del nivel del equipo (figura 3.8). En la figura 3.4 podemos ver los 10 mejores ratings que quedarían en el dataset tras el escalado.

Después de este tratamiento, se ha insertado nuestra variable objetivo, que será el **Rating del mismo jugador el año siguiente**, además del nombre del equipo y su puntuación, que dan más contexto de su situación futura, pero solo con datos que están disponibles de esa misma temporada.

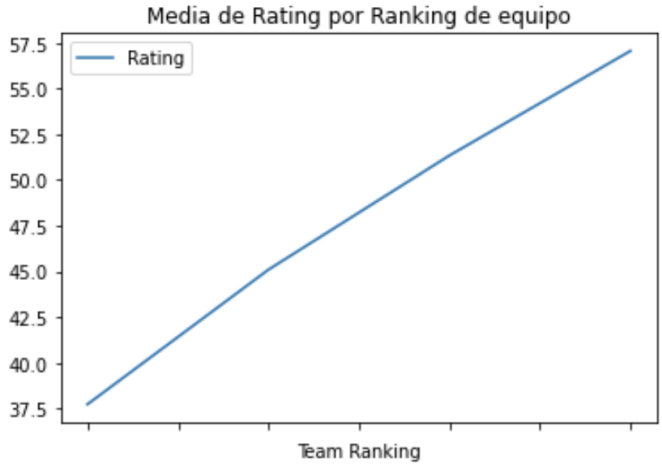


Figura 3.8: Valor medio de Rating en función de la puntuación de equipo

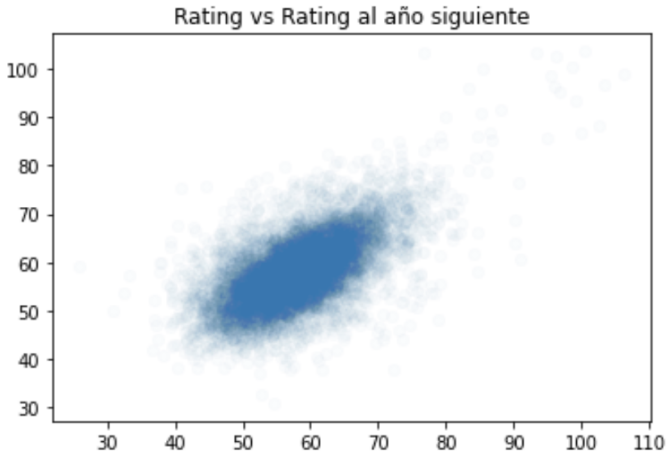


Figura 3.9: Diagrama de dispersión que compara el Rating actual(eje x) frente al Rating al año siguiente (eje y)

4

ALGORITMOS

En esta sección se va a hablar de la parte relacionada con la inteligencia artificial de nuestro sistema, es decir de los algoritmos de machine learning aplicados para predecir la variable y descrita.

Este tipo de problema está en el mundo de los problemas de regresión, puesto que lo que se quiere predecir es una variable continua.

El proceso que se ha seguido es el siguiente:

1. Se ha realizado una optimización de los hiperparámetros de los algoritmos mediante *Grid Search* con validación cruzada.

2. Después de obtener los mejores parámetros, se ha procedido a entrenar los modelos utilizando validación cruzada en 10 iteraciones (o 10K Fold en inglés) 5 veces y promediado sus métricas.

3. Para medir la capacidad de predicción de los algoritmos hemos utilizado tres métricas ampliamente utilizadas para evaluar el rendimiento en problemas de regresión: Error absoluto medio (MAE), Raíz del error cuadrático medio (RMSE) y el coeficiente de determinación.

Todos estos algoritmos han sido probados en un entorno *miniconda*, donde se ha hecho uso del lenguaje *python3* y de las siguientes librerías disponibles, que contienen todos los algoritmos que se van a utilizar.

Scikit Learn: Librería enfocada a aprendizaje automático que ofrece un amplio catálogo de algoritmos de regresión para el proyecto, así como métodos de validación, preprocesamiento de datos y selección de los hiperparámetros para nuestros algoritmos.

XGBoost: Librería destinada al uso de distintos algoritmos de aprendizaje automático basados en el concepto del descenso por gradiente extremo.

Ahora se hará una explicación en líneas generales cómo funcionan los 3 algoritmos que mejores resultados han dado, que son **Elastic Net** [40], **Random Forest** [19] y **XGBoost** [4]. Pero antes, se considera conveniente hablar del método que se ha seguido para realizar el ajuste hiperparámetros.

4.1. Grid Search

Esta técnica de optimización de hiperparámetros hace una búsqueda exhaustiva en base a los valores candidatos de los hiperparámetros que se desee analizar.

Esta búsqueda lo hace generando un modelo por cada una de las opciones posibles que genera el producto cartesiano de los posibles valores de los hiperparámetros que se desee probar, entrena cada modelo con la métrica de elección (en este caso, MAE con validación cruzada en 10 iteraciones) y al final obtiene los hiperparámetros del modelo que mejor se ajusta a los datos.

Esta técnica tiene sus desventajas de cara a la dimensionalidad de las pruebas que se quieran ejecutar sobre el algoritmo, porque rápidamente sus tiempos de ejecución pueden disparase si se decide probar con algunos parámetros más. Es por ello que requiere de bastante tiempo de ejecución en líneas generales, aunque debido a que la mayoría de los hiperparámetros de los modelos son independientes unos de otros es una técnica altamente paralelizable.

El proyecto ha hecho uso de la clase **GridSearchCV**, de la libreria de **Scikit Learn**.

4.2. Elastic Net

En este caso, este modelo está dentro de los llamados modelos lineales, que pretenden explicar el comportamiento de una variable como una función lineal que involucre a las variables predictoras.

Esta labor, sin embargo, se ve fuertemente influenciada por la cantidad de variables que estén presentes en la función y como estas se correlacionan unas con otras. Este tipo de hechos pueden hacer que un modelo lineal normal y corriente sea poco fiable a la hora de establecer predicciones.

Es aquí donde se introduce el concepto de regularización, cuyo fin es básicamente mejorar el rendimiento de la regresión a base de reducir la complejidad del modelo. Esto lo consiguen integrando a la función de coste una penalización adicional que aumenta a más complejo sea el modelo.

A continuación se van explicar las dos técnicas de regularización que serán relevantes al uso de este algoritmo.

Regularización Ridge *L2*

Este tipo de regularización pretende minimizar del modelo coeficientes muy grandes en algunas de las variables predictoras. Esto lo consigue añadiendo a la función de coste L la suma de los cuadrados de los coeficientes de las variables (ω). Debido a que esta suma puede alcanzar valores muy grandes, se añade una constante (λ) que reduce o aumenta el grado de penalización segun su valor, siendo 0 una penalización nula y por tanto obtenemos un modelo lineal por mínimos cuadrados ordinarios u

OLS.

La fórmula quedaría de la siguiente manera:

$$L_2 = L(X, \omega) + \lambda \sum \omega_i^2$$

Esto permite reducir la varianza del modelo y, por tanto, mejorar la precisión del mismo. El principal problema de este tipo de regularización es que, pese a que los coeficientes de las variables menos relevantes tenderán a 0, nunca llegarán a 0. Este problema no afectará demasiado a la precisión del modelo puesto que tendrían valores sumamente pequeños, pero hacen más difícil la interpretación del mismo.

Regularización Lasso *L1*

Este tipo de regularización pretende minimizar del modelo coeficientes que sean irrelevantes de algunas de las variables predictoras. Esto lo consigue añadiendo a la función de coste L la suma de los valores absolutos de los coeficientes de las variables (ω). Del mismo modo que la anterior técnica, se añade una constante (λ) que reduce o aumenta el grado de penalización según su valor, siendo 0 una penalización nula y por tanto obtenemos un modelo lineal por mínimos cuadrados ordinarios u ***OLS***.

La fórmula quedaría de la siguiente manera:

$$L_1 = L(X, \omega) + \lambda \sum |\omega_i|$$

La regularización Lasso también reduce los valores de los coeficientes. Sin embargo, en este caso sí que puede darse que los valores sean 0. Esto significa que esta regularización es muy útil para descartar variables predictoras que no sean útiles de cara a la regresión, con la desventaja de que esto puede llegar a ser en ciertos casos extremo y se puede ver perjudicado el rendimiento del modelo lineal, por lo que la elección del valor de λ en este caso tiene mayor importancia que con Ridge.

Elastic Net

Ambas técnicas tienen tanto ventajas como desventajas. Elastic Net surge como una alternativa que pretende incluir lo mejor de ambas. La idea principal de Elastic Net consiste en utilizar en aliviar esas desventajas que puede tener Lasso, por lo que incorpora a su función de coste cierta estabilización con el concepto de regularización de Ridge. Estas no se aplican de forma completa, sino que las relaciona linealmente.

La función de coste L se ve modificada con una penalización también, pero esta surge como la suma de los dos tipos de penalización, reguladas por un valor α definido el intervalo [0,1] que indicará

cuánto de cada regularización se aplica a la función de coste.

La función de coste de Elastic Net es la siguiente:

$$\frac{||y - X\omega||^2}{2 * n_{samples}} + \lambda * \alpha * ||\omega|| + 0{,}5 * \lambda * (1 - \alpha) * ||\omega||^2$$

En esta función de coste, un valor de α de 1 equivaldría a una regularización Lasso estándar. Como interesa descartar variables del modelo que no aporten información sobre la predicción, es muy común ver en las aplicaciones de este algoritmo valores de α bastante altos, y solo dejando un pequeño margen para Ridge que aporte una estabilidad al modelo.

Pero antes de proceder con la implementación en concreto del algoritmo, es importante hablar de una de las características más importantes de cara a los datos que se le van a introducir al algoritmo, que es la escala de los mismos.

Estandarización de atributos

Debido a que este tipo de modelos establecen una relación lineal entre la variables independientes y la variable dependiente, es de vital importancia manejar qué rangos de valor tienen los datos. Una diferencia muy grande entre los rangos de valor de una columna y otra puede decantar el modelo a favorecer de forma inapropiada a la variable que se mueve en rangos de valor más altos. Del mismo modo, una variable que varía poco en sus valores hará que las pequeñas diferencias que pueda haber se vean poco apreciadas por el modelo, aun cuando puedan ser de vital importancia. En este caso es algo evidente, puesto que se manejan algunas variables que se mueven en rangos mucho mayores que otros, como puede ser la puntuación de equipo (Valores de entre 0 y 19000 con mucha varianza) y el Rating (valores de 0 a 10, con bastante concentración en algunos puntos)

Es por ello que se debe realizar un preprocesado de los datos antes de entrenar al modelo con ellos. Para ello lo que se hará es centrar las variables, que consiste en restar a los datos de una columna la media de la misma, y por último reducir el rango de las variables, dividiendo los datos centrados por la desviación típica de la distribución.

El escalado entonces se realiza de la siguiente manera, cuya implementación se encuentra en la clase StandardScaler [41], de la librería de Scikit Learn:

$$z = \frac{x - \mu}{\sigma}$$

Aclarados estos conceptos, terminamos con la implementación del modelo.

Optimización de hiperparámetros

Dentro de los recursos mencionados, la librería de Scikit Learn incorpora la clase ElasticNet [42], que incorpora el uso de este algoritmo.

Se ha creído conveniente realizar pruebas con los siguientes parámetros:

alpha: Referido a la constante de regularización de la penalización aplicada, en nuestras fórmulas equivale a λ.

l1_ratio: Referido a los posibles valores que puede tener la proporción de cada regularización, que equivale en la formula a los posibles valores de α.

max_iter: Referido a las épocas de entrenamiento de nuestro algoritmo.

Código 4.1: Fragmento de código de la prueba de hiperparámetros en Elastic Net

```
param_grid = {
    "alpha": [0.001, 0.01, 0.1],
    "l1_ratio": np.arange(0.1,1,0.01),
    "max_iter": [1000, 5000, 10000],
}
cv = KFold(n_splits=10, shuffle=True, random_state=1)
el = GridSearchCV(ElasticNet(fit_intercept=True), param_grid=param_grid, cv=cv,
      scoring='neg_mean_absolute_error')
```

Tras la prueba se obtiene que los resultados que mejor han funcionado en el modelo son 'alpha': 0.001, 'l1_ratio': 0.99, 'max_iter': 1000

4.3. Random Forest

Random Forest es un algoritmo de aprendizaje automático que está basado en el concepto de los árboles de decisión.

Árboles de decisión

Los árboles de decisión son un tipo de modelo que, dado un conjunto de datos, busca hacer una clasificación efectiva de los mismos mediante la creación de un sistema de reglas jerarquizadas en forma de árbol. Cada fila pasa por las ramas correspondientes a las reglas que cumple, llegando a una clasificación final al llegar a las hojas del árbol. Este tipo de modelos gozan de una gran interpretabilidad. Al tener un sistema de reglas claro, este permite tener una representación muy clara del razonamiento del modelo a la hora de clasificar.

Otras ventajas que ofrecen este tipo de modelos es que sirven tanto para la clasficación de variables

con valores discretos como para la regresión, que es el tipo de problema que se está tratando ahora mismo. Además, no es necesario hacer un escalado de los datos al no asumir que hay una relación lineal entre las variables predictoras y la variable a predecir.

Sin embargo, este tipo de modelos deben ser tratados con cierto cuidado a la hora de configurarse, puesto que tienen especial tendencia a sobreajustarse, es decir, entrenarse demasiado y solo conocer el dominio de los datos de entrenamiento sin capacidad de generalizar más allá de lo que ha sido alimentado el modelo.

Bagging

Como mejora a esto, se propone un tipo de modelo resultante de la combinación de varios de estos árboles. Este tipo de métodos se conocen como **agregación de bootstrap** o *bagging* en inglés. Este tipo de modelos buscan reducir la tendencia de los originales a sobreentrenarse mediante la creación de más de un árbol de entrenamiento.

El proceso llevado a cabo por esta técnica es muy sencillo:

1.– Se dividen los datos de forma aleatoria en un número N de subconjuntos

2.– Por cada uno de los subconjuntos, se genera un árbol de decisión

3.– A la hora de clasificar, se pasan los datos de forma independiente a cada uno de los árboles generados. El resultado final del modelo será la clase más elegida para problemas de clasificación y la media de todos los árboles para regresión.

Esto genera un mejor rendimiento, pero a costa de perder interpretabilidad del modelo.

Random Forest: Feature Bagging

Por último, esto lleva al tipo de modelo que es el Random Forest. Este tipo de modelos pierden casi la totalidad de la característica que en primera instancia hacía muy atractivo a los árboles de decisión, que era su facilidad a la hora de ser interpretado por un ser humano. Sin embargo, este modelo genera un rendimiento excelente en una cantidad enorme de problemas sin necesidad de un tiempo extenso de ajuste de hiperparámetros.

Este tipo de modelo bebe de los conceptos que se han explicado anteriormente, con la diferencia de que Random Forest no hace un *bagging* de los datos con los que se va a entrenar, sino que realizan lo que se conoce como *feature bagging*. Es decir, lo que se están generado son distintos árboles de decisión, no entrenados sobre distintas filas de datos, sino sobre distintas columnas de una misma fila.

El proceso sería el siguiente:

1.– Se selecciona un número k de variables elegidas aleatoriamente

2.– Se elabora un árbol de decisión con esas variables

3.– Se repiten los pasos 1. y 2. hasta que se obtenga el número de árboles que necesitemos

4.– A la hora de clasificar, se pasan los datos de forma independiente a cada uno de los árboles generados. El resultado final del modelo será la clase más elegida para problemas de clasificación y la media de todos los árboles para regresión.

Este algoritmo permite trabajar con un número grande de variables y dar estimaciones sobre qué variables son las más relevantes a la hora de la regresión.

Optimización de hiperparámetros

La librería **Scikit Learn** implementa la clase **RandomForestRegressor**, que nos permite generar un modelo de regresión con este algoritmo.

Para este caso se ha decidido optimizar los siguientes hiperparámetros

n_estimators: Referido al número de árboles generados por el modelo.

max_features: Referido al número de atributos en los que mirar para generar la mejor division

max_depth: Referido a la profundidad máxima que puede alcanzar un árbol

Código 4.2: Fragmento de código de la prueba de hiperparámetros en Random Forest

```
param_grid = {
    'n_estimators': [200, 500],
    'max_features': ['auto', 'sqrt', 'log2'],
    'max_depth' : [5,6,7,8],
}
gscv = GridSearchCV(RandomForestRegressor(), param_grid=param_grid, cv=cv,
     scoring='neg_mean_absolute_error', n_jobs=-1)
```

Tras la prueba obtenemos que los resultados que mejor han funcionado en el modelo son 'max_depth': 8, 'max_features': 'auto', 'n_estimators': 500

4.4. XGBoost

El nombre viene dado por las siglas de ***eXtreme Gradient Boosting***. Es un algoritmo basado en el concepto de árboles de decisión reforzados mediante la optimización por gradiente, lo que se conoce como *Gradient Boosting*. Este algoritmo está ganando mucha popularidad recientemente en plataformas de competición de Data Science y Machine Learning como Kaggle, debido a la potencia de predicción y el rendimiento a nivel computacional que tiene a la hora de entrenarse. Antes de seguir hablando de las mejoras que ofrece este algoritmo, es importante hacer una explicación del concepto sobre el que está basado.

Gradient Boosting

El concepto detrás de todas las variantes de este paradigma es exactamente el mismo, reducir el error del modelo en base a la creación de modelos llamados *weak learners* que iterativamente van buscando reducir el residuo (la diferencia entre la predicción del modelo y el valor real).

Esto se realiza de la siguiente manera:

1. Se ajusta un primer *weak learner*, llamado $f_1(X)$ que intenta predecir la variable dependiente y:

$$f_1(X) \approx y$$

2. Se calcula el residuo de la primera iteración ($f_1(X) - y$) y se genera un nuevo *weak learner*, al que llamaremos $f_2(X)$, que en este caso lo que hará será intentar predecir el residuo de la primera iteración:

$$f_2(X) \approx f_1(X) - y$$

3. Se calcularía el residuo de esa iteración y se generaría un nuevo *weak learner* que intentaría predecir el residuo, y así sucesivamente hasta generar M modelos que han ido sucesivamente corrigiendo los errores de las anteriores iteraciones con una constante de aprendizaje λ para ayudar a prevenir el sobreajuste. Obteniendo al final la siguiente predicción:

$$y \approx \lambda f_1(X) + \lambda f_2(X) + \lambda f_3(X) + ... + \lambda f_M(X)$$

Al estar introduciendo nuevos modelos que buscan iterativamente reducir el error de los modelos anteriores, está realizando un descenso por gradiente, de ahí su nombre.

Qué mejora *XGBoost*

Como se ha mencionado, *XGBoost* está basando en el concepto anterior, sin embargo, este trae algunas mejoras, tanto a nivel algorítmico como computacional que incrementan el rendimiento de estos modelos y que han convertido este algoritmo en uno de los más efectivos a día de hoy y en uno de los principales candidatos a considerar.

- Soporte para la paralelización, que incrementa significativamente el rendimiento, soportando además el uso de GPUs.

- Poda en base a profundidad, con el parámetro **max_depth**

- Optimización del rendimiento adicional a nivel de hardware.

- Incorporación de regularización *LASSO* y *Ridge*, que penalizan la complejidad del modelo

y previene el sobreajuste.

- Admisión de variables con alta dispersión, encontrando el mejor valor no presente.

- Puntos de separación en los datos mediante la búsqueda de cuartiles óptimos.

- Validación cruzada como método de validación por defecto.

Optimización de hiperparámetros

En la librería XGBoost hay a disposición un algoritmo de regresión con este concepto llamado **XGBRegressor**.

Sobre este algoritmo se han decidido ajustar los siguientes parámetros:

min_child_weight: Referido a la suma mínima que debe tener cada árbol hijo en la instancia.

gamma: Referido a la reducción mínima del error que debe producirse en un árbol para generar una hoja nueva.

subsample: Referido al porcentaje del dataset que se usará en la generación del modelo.

colsample_bytree: Referido al porcentaje del columnas del dataset que se usará en cada instancia.

max_depth: Referido a la profundidad máxima que puede alcanzar un árbol

Código 4.3: Fragmento de código de la prueba de hiperparámetros en XGboost

```
param_grid = {
        'min_child_weight': [1, 5, 10],
        'gamma': [0.5, 1, 1.5, 2],
        'subsample': [0.6, 0.8, 1.0],
        'colsample_bytree': [0.6, 0.8, 1.0],
        'max_depth': [4, 5]
        }
gscv = GridSearchCV(XGBRegressor(), param_grid=param_grid, cv=cv,
      scoring='neg_mean_absolute_error', n_jobs=-1)
```

Tras la prueba se obtiene que los resultados que mejor han funcionado en el modelo son 'colsample_bytree': 0.8, 'gamma': 2, 'max_depth': 4, 'min_child_weight': 5, 'subsample': 1.0

4.5. Resultados

En esta sección se mostrarán los resultados que ha obtenido el modelo y se discutirán de cara a sacar conclusiones sobre las hipótesis planteadas a lo largo del proyecto.

Las métricas con las que se evaluarán los modelos son las siguientes:

Error Absoluto Medio (MAE): Mide el error tomando el valor absoluto de la resta del valor predicho menos el valor real para todos los ejemplos de la fase y hace la media de los mismos.

Raiz del Error Cuadrático Medio (RMSE): Mide el error tomando el cuadrado de la diferencia entre el valor predicho menos el valor real, calcula la raiz de ese resultado y toma la media de todos los ejemplos. Se diferencia del primero en que este es más sensible a las grandes diferencias entre lo predicho y lo real.

Puntuación R^2: Medida que permite ver como es el modelo capaz de replicar los resultados, calculado como el cuadrado del coeficiente de correlación de Pearson.

Los resultados para estos 3 algoritmos son los siguientes:

Modelo	MAE	RMSE	R^2
ElasticNet	0.52	0.67	0.55
Random Forest	3.64	4.76	0.54
XGBoost	3.67	4.81	0.53

Tabla 4.1: Tabla de resultados

4.5.1. Discusión

Estos resultados requieren de un contexto para entenderse mejor, puesto que tanto MAE como RMSE dependen del rango de la variable, y es preciso recordar que Elastic Net ha sido alimentado con datos escalados, por lo que no comparten el mismo rango que los demás.

Como se puede comprobar, los 3 modelos presentan resultados similares, aunque en el contexto del rango de las variables parece que Random Forest y XGBoost tienen un menor MAE y RMSE que Elastic Net, puesto que el rango de valores de estos es mucho mayor que en el caso del último. Sin embargo, en cualquier caso, son resultados que si bien son prometedores, su precisión tiene un rango de mejora importante, sobre todo los R^2, que no terminan de ser del todo certeros.

Esto puede deberse a diversos motivos. La falta de datos adicionales debido a la tan estricta estructura del dataset (requiriendo cada temporada de un jugador de disponer de información completa de la siguiente, que en muchísimos casos no es así) puede ser uno de los principales factores que ha influido en esta diferencia de resultados, sino el principal. Una nueva extracción en la que no solo se tengan los datos de los jugadores actuales, sino un histórico de los jugadores que hayan jugado en todos estos equipos, en el que se pueda contar con jugadores que ya se han retirado, o jugadores que han realizado carreras en estos equipos y que ahora se encuentran en equipos de ligas menores...

Claramente el número de filas disponibles aumentaría de forma considerable, aunque el proceso de extracción de estos nuevos datos sería un proceso bastante más largo y complicado que el realizado aquí.

Un problema principal que ha podido influir en el resultado es la presencia de valores y muy atípicos en la distribución. Estos valores incluyen a las mejores temporadas que figuran en nuestro dataset, que mientras que la mayor parte de valores se concentran en torno a un rango de valores, el escalado de valores planteado supone la existencia de valores que se separan de la distribución, por tanto al ser predichas incurren en un error muy grande.

Por último, una ultima teoría que podría haber supuesto un impedimento para conseguir mejores resultados es el no tener en cuenta la posición del jugador para determinar la influencia de ciertas estadísticas en su Rating. En caso de haber tenido más filas se hubiera realizado un modelo por posición para especializarse en qué atributos hacen que el Rating aumente o disminuya en la posición. Es posible que debido a la naturaleza caótica del método de *web scraping* haya habido cambios entre y durante extracciones que hayan generado datos anómalos en nuestro dataset, sin embargo a excepción de unas pocas filas no parece ser el caso, aunque es una tarea pendiente como trabajo futuro.

4.5.2. Influencia de fuentes adicionales y escalado

Una de las premisas que motivaron la creación del dataset y la búsqueda de fuentes adicionales era la idea de que valorar el rendimiento de un jugador tendría más sentido si se analiza en el contexto táctico y el nivel del equipo en el que esté (ver 2.1).

Para comprobar esa teoría, se ha realizado un experimento donde no figuran los datos de las fuentes de puntuación de equipo ni los datos del equipo. Esto implica que las variables de Rating no han escalado según el método que se ha comentado en la sección 3.5.

En el experimento se ha seguido el mismo procedimiento que para la idea base. Debido a la naturaleza de las métricas de error absoluto medio y raíz de error cuadrático medio, que dependen del rango de valores que tenga la variable, vamos a compararlo en función de la métrica R^2.

Los resultados obtenidos en el experimento son los siguientes:

Modelo	MAE	RMSE	R^2
ElasticNet	0.67	0.87	0.24
Random Forest	0.23	0.30	0.24
XGBoost	0.23	0.30	0.26

Tabla 4.2: Tabla de resultados sin escalado ni fuentes adicionales

Como se puede ver, los valores de R^2 para este experimento son considerablemente más bajos que en nuestra hipótesis, lo cual refuerza la idea de que el contexto y el nivel del club en el que el jugador está desarrollando sus habilidades parece afectar claramente a sus valores de rendimiento (al menos, con la medida Rating que se está usando).

4.5.3. Predicción de rendimiento en jugadores prometedores

Una parte donde el proyecto parece haber presentado un rendimiento prometedor es a la hora de predecir grandes saltos en el rendimiento de algunos jugadores. Un ejemplo de ello es el salto en el rendimiento que tuvo el alemán Serge Gnabry en la 2018/2019 al terminar su cesión en el TSG 1889 Hoffenheim y recalar en el Bayern de Münich, ambos de la Bundesliga de Alemania. En la temporada anterior su Rating escalado estaba en 43.09, el modelo estimó un crecimiento muy grande y estimó un rendimiento la temporada siguiente de 58.69, que se acerca bastante al rendimiento de 62.48 que tuvo esa temporada en el conjunto bávaro, por lo que el proyecto, en su estado actual, podría ser prometedor para predecir crecimientos de forma grandes en jugadores en sus próxima campañas. Esto es un ejemplo de la robustez que presentaría el proyecto por la zona donde están más concentrados los datos, que sin duda es una noticia prometedora.

5

Herramienta de recomendación e insights

En este capítulo se va a analizar, según el modelo, uno de los fichajes mas mediáticamente cubiertos e influyentes de la historia reciente.

El caso de estudio

El caso a estudiar es el traspaso de Neymar da Silva Santos Júnior, más conocido simplemente como Neymar. Jugador importante del Fútbol Club Barcelona de Primera División de España durante los años 2013 y 2017, fue traspasado al Paris Saint-Germain de la Ligue 1 de Francia por la astronómica cifra de 222 millones de euros que tenía el jugador como clausula de rescisión, convirtiéndose en el fichaje más caro de la historia del fútbol a día 13 de junio de 2021.

La idea principal es, en base a los datos disponibles y también a los modelos de inteligencia artificial, estudiar de una forma más analítica el por qué de ese fichaje.

Este estudio no va a tomar en cuenta el precio pagado por el jugador, puesto que no está en el dominio del problema y simplemente se busca analizar el valor futbolístico que tiene el jugador que aportar al que iba a ser su nuevo club.

5.1. Análisis del jugador

Neymar es un jugador brasileño que se desarrolla principalmente en la demarcación de extremo izquierdo (jugador de ataque que suele desarrollar sus acciones en el lado izquierdo del tercio rival de campo), aunque su versatilidad con ambas piernas y sus cualidades técnicas le permiten jugar de otras posiciones como en la banda contraria o de delantero centro. Desde ahí es un jugador capaz de generar un número importante de goles, pero también de asistencias, además de ser un jugador que genera ventajas con el regate, donde es considerado uno de los mejores jugadores del planeta. (ver figura 5.1).

Season	Team	Goals	Assists	PS%	DrB (off)
2015/2016	Barcelona	27.0	16.0	0.814461	189.0
2014/2015	Barcelona	32.0	7.0	0.813454	126.0
2016/2017	Barcelona	17.0	19.0	0.761072	215.0
2013/2014	Barcelona	13.0	11.0	0.851393	94.0
2017/2018	PSG	25.0	16.0	0.792139	192.0
2018/2019	PSG	20.0	9.0	0.815533	102.0
2019/2020	PSG	16.0	10.0	0.790080	128.0

Figura 5.1: Estadísticas importantes de Neymar por temporada

5.2. Barcelona vs PSG

Vamos a estudiar cómo jugaban estos dos equipos durante la temporada 2016/2017. Hemos tomado las estadísticas de ambos equipos y hemos decidido hacer una comparativa de algunos de los aspectos ofensivos principales. Algunas de las estadísticas que comparamos figuran en el anexo C, sin embargo, para comprobar cuánto se parecen ambos equipos, vamos a hacer uso de la herramienta de Graphext [43]. Graphext es una startup española que desarrolla una plataforma web que permite realizar análisis de datos y proyectos de data science de forma visual, a los que agradecemos por su amable trato y por ofrecer la infraestructura que ha hecho posibles estas figuras.

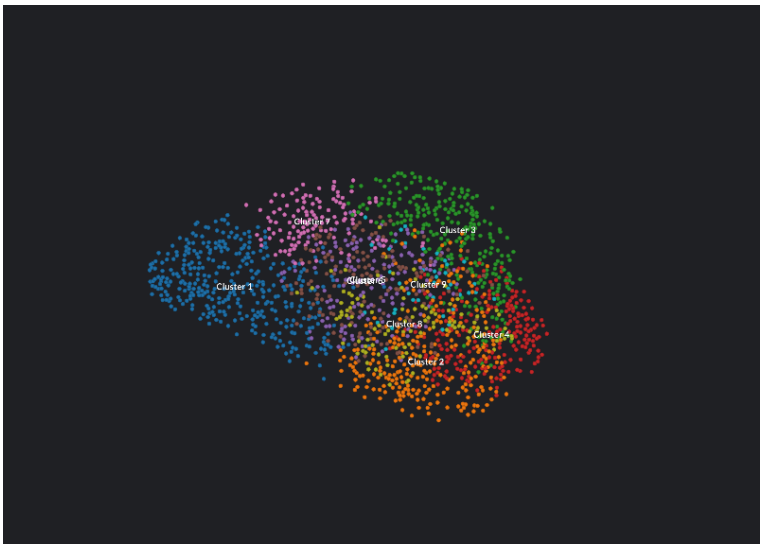


Figura 5.2: Visual de los datos de equipo agrupados en clusters

Como podemos comprobar en las figuras 5.2 y 5.3, donde figuran todos los equipos de todas las temporadas que hemos recogido y realiza un proceso de clusterización para agrupar equipos con características similares, no solo el algoritmo de clusterización que utiliza la herramienta los coloca en

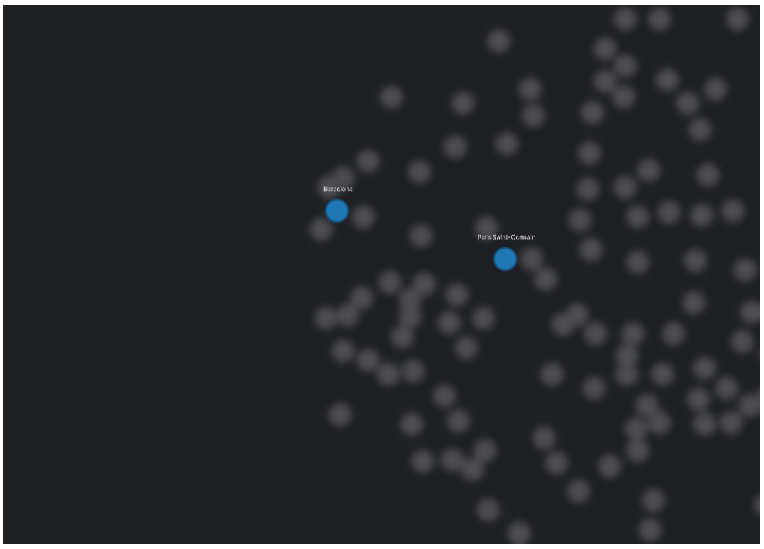


Figura 5.3: Sección del grafo con la posición de Barcelona y PSG en la temporada 2016/2017

el mismo cluster, sino que además están ambos entre sus vecinos más cercanos. Más en concreto, PSG es el segundo vecino más cercano al Barcelona en la temporada 2016/2017), lo que indicaría que tienen estilos de juego similares. Sin embargo, además de la cantidad de goles que tuvieron los dos equipos (116 Barcelona vs 83 Paris Saint-Germain) hay un par de diferencias que queremos resaltar entre los dos equipos que son importantes de cara a analizar este fichaje.

El Barcelona realiza 2.2 más regates por partido, lo que podría explicar cosas como que rematen más por la zona de penalti, que reciban más faltas y, sobre todo, que generen un mayor porcentaje de goles desde jugadas (vease *Open Play*), donde un regate permite generar desequilibrio en las defensas.

Ya desde aquí se puede ver que el PSG podría verse en la situación de buscar un jugador con las características de Neymar. El equipo podría estar interesado en incrementar y/o hacer más efectivas sus acciones en ataque por banda izquierda, o buscar un jugador desequilibrante con regates que genere ventaja y facilite la creación de goles, bien disparando o asistiendo en ellos.

5.3. Neymar vs PSG

Una vez hecha esa observación sobre los estilos de juego de ambos equipos, en los que se ha visto que el PSG podría verse interesado en buscar un regateador goleador, se van a comparar los números de Neymar frente al resto del jugadores del PSG en la posición de extremo izquierdo.

Es evidente a simple vista que la producción goleadora es muy superior al resto de los jugadores del Paris Saint-Germain, siendo superior a las dos mejores opciones que parece tener equipo de cara a atacantes por la banda izquierda en goles y destacándose como un jugador excelente en el último

	Player Name	Season	Team	Goals	Assists	PS%	DrB (off)	Rating
526	Neymar	2016/2017	Barcelona	17.0	19.0	0.761072	215.0	96.264083
2019	Lucas Moura	2016/2017	PSG	13.0	6.0	0.822858	60.0	71.471852
3456	Angel Di Maria	2016/2017	PSG	10.0	8.0	0.772975	34.0	69.161658
6464	Layvin Kurzawa	2016/2017	PSG	2.0	5.0	0.872272	31.0	67.163454
9794	Julian Draxler	2016/2017	PSG	5.0	1.0	0.829270	24.0	67.663186
13999	Christopher Nkunku	2016/2017	PSG	1.0	0.0	0.852971	2.0	53.033211

Figura 5.4: Comparativa Neymar vs PSG

pase con respecto al resto de jugadores. Parece ser peor respecto al porcentaje de pases en general, pero sin embargo donde no parece tener competición es con respecto a los regates, donde está es muchísimo más prolífico que el resto.

Debido a la diferencia de puntuaciones de equio, su Rating se ve beneficiado, pero aun así la diferencia es bastante grande con respecto a las opciones principales del equipo.

Neymar se postula como una opción superior en múltiples aspectos al resto de jugadores que juegan por banda izquierda.

5.4. Predicción de rating

Todo parece indicar que Neymar es un fichaje más que deseable para el club, comparando la producción estadística y la relevancia que puede tener el jugador en el club en el que actualmente se encuentra jugando. Sin embargo hay una cosa que todavía no sabemos y es cómo rendirán estos jugadores si se mantuviese es *status quo* en sus carreras y siguieran con sus equipos.

Para ello hemos decidido tomar los dos principales competidores de Neymar en este equipo, que son Ángel Di María, de Argentina y Lucas Moura, de Brasil. Vamos a comparar los ratings que predice el modelo, extrayendolos de la fase de entrenamiento y haciendo una predicción sobre ellos. Es esperable que el modelo sea muy conservador con el rendimiento de Neymar, puesto que es una temporada cuyo Rating figura undécima entre todas las temporadas de jugadores que hay, siendo un valor atípico o *outlier*.

Jugador	Rating 16/17	Predicción 17/18	Rating 17/18
Neymar	96.26	78.23	102.5
Lucas Moura	71.47	69.71	54.80
Di Maria	69.16	70.17	67.89

Tabla 5.1: Tabla de predicciones de los jugadores mencionados para la siguiente temporada

Si bien son datos que son claramente conservadores por parte de Neymar debido a lo comentado recientemente, sigue valorando a este jugador con bastante holgura del resto, poniendo 8 puntos de diferencia del mejor caso.

El resultado a la siguiente temporada es que Neymar protagonizó la quinta mejor temporada del dataset, llevando su producción a un nivel aun mayor en este equipo. Mientras tanto, Di María mantuvo una producción similar y Lucas Moura vio su Rating muy perjudicado, debido al cambio de equipo y pocos minutos de juego de los que dispuso.

Player Name	Season	Team	Goals	Assists	PS%	DrB (off)	Rating
Neymar	2017/2018	PSG	25.0	16.0	0.792139	192.0	102.496264
Angel Di Maria	2017/2018	PSG	12.0	7.0	0.793261	40.0	67.894986
Layvin Kurzawa	2017/2018	PSG	5.0	3.0	0.869173	20.0	66.233737
Julian Draxler	2017/2018	PSG	4.0	6.0	0.926836	31.0	61.719698
Yuri Berchiche	2017/2018	PSG	2.0	4.0	0.891198	12.0	64.439851
Christopher Nkunku	2017/2018	PSG	4.0	0.0	0.922000	5.0	53.891570
Lucas Moura	2017/2018	Tottenham	0.0	1.0	0.779000	2.0	54.804679

Figura 5.5: Estadísticas de jugadores en banda izquierda PSG + jugadores analizados en 2017/2018

En conclusión, pese a los resultados anómalos que supone para el modelo la situación de un jugador de las características de Neymar, se puede ver, junto a un análisis de los datos que indican alguno de los aspectos tácticos, que el Paris Saint-Germain podría querer reforzarse en la posición que cubre el jugador de nuestro estudio que el fichaje de Neymar constituye una mejora en los resultados del equipo.

Con la intención de compartir esto con aficionados y expertos en fútbol y/o data science, en un futuro se publicarán todas herramientas de datos en el repositorio público de Github, en el momento de la publicación del trabajo se encuentra en una fase temprana.

6

Conclusiones y trabajo futuro

6.1. Conclusiones

Este trabajo de fin de grado hace hincapié en conceptos cruciales de la ciencia de datos. Empezando por la investigación de trabajos previos relacionados con el mundo de los deportes, con el objetivo de establecer un estado del arte lo más desarrollado posible. En base a algunas aportaciones que se ha visto que se podían hacer al dominio, se ha hecho una búsqueda de fuentes que fueran lo más completas y detalladas posible dados los recursos que estaban al alcance de un trabajo de estas magnitudes.

Al final se decidió tomar las fuentes de Club World Ranking para tomar un valor que indique la fortaleza del equipo a nivel global y WhoScored porque ofrece un desglose estadístico de jugadores y equipos de forma estandarizada, además de contener nuestra variable de predicción, que está relacionado con la variable Rating que presenta la propia fuente. En el caso de este proyecto, se buscaba tanto el Rating del jugador en una temporada como en la siguiente.

En esa extracción se ha profundizado en el concepto del web scraping y se han desarrollado, mediante las herramientas de Selenium y Beautiful Soup, una serie de scripts para extraer los datos que en su momento se creían convenientes para el proyecto. Una vez hecha la extracción se ha hecho un tratamiento de valores nulos y estandarización de valores en algunas columnas dependiendo del dataset y se ha hecho una fusión de las 3 fuentes que da lugar a un dataset de 17589 filas y 97 columnas.

Una vez formado el dataset, hemos realizado un escalado de valores de las variables relacionadas con el Rating en base a la hipótesis principal del proyecto, que era que el rendimiento de un jugador en la siguiente temporada podría ser estimado con mayor precisión si se pone en contexto las estadísticas de un jugador con las estadísticas y el nivel global del club en el que juega esa temporada.

A partir de ahí se ha profundizado en aspectos del aprendizaje automático, estudiando los algoritmos propuestos para el experimento, además del uso de técnicas como la optimización de hiperparámetros, la regularización del aprendizaje y la escalado de datos para el aprendizaje.

El modelo resultante parece confirmar la hipótesis de que el contexto estadístico y táctico del equipo y el nivel general del mismo favorece el estudio de la progresión de un jugador en lo que respecta a su rendimiento las próximas temporadas.

En base a los datos disponibles y los modelos generados, se ha realizado una aplicación práctica del proyecto, intentando explicar el por qué del fichaje del jugador Neymar al Paris Saint-Germain francés en la temporada 2017/2018, donde concluimos que el fichaje claramente hubiera beneficiado al equipo, pero sin estimar del todo bien su rendimiento debido a presentar valores atípicos, sobre los que se pretende trabajar en un futuro.

6.2. Trabajo futuro

En la fase de análisis de resultados (ver 4.5) se han comentado algunos de los problemas más importantes que presenta este trabajo. Estos posiblemente tengan una influencia muy grande en el resultado final, pero el proyecto constituye en su estado actual una linea base sobre la que trabajar en futuras iteraciones, posiblemente como trabajo de fin de máster.

Uno de los problemas más relevantes es el hecho de que se recoge solamente un histórico de los jugadores actuales, descartando una fracción importante de jugadores que han pasado por las plantillas de las que se han recogido sus datos. De esa forma, un mayor número de jugadores potenciales para estudio estarán a disposición de los usuarios y será probablemente la primera mejora que se llevara a cabo.

Otra mejora que aumentaría la calidad de la herramienta que se propone es que a día de hoy no es fácilmente utilizable, por lo que el podría ser interesante, una vez el dataset goce de una mayor calidad, usar herramientas de visualización a la hora de buscar posibles fichajes con todos los datos que disponemos.

Respecto a los modelos utilizados, si bien se han probado más algoritmos que no se han comentado debido a sus malos resultados, se estudiarían más modelos o enfoques. Uno de los más interesantes está relacionado con el uso de series temporales que entrenen a redes LSTM para hacer un análisis de un jugador en base a su progreso a lo largo de los años y así obtener resultados en base a las tendencias del jugador, lo que lo convierte en una opción muy interesante de cara a la idea inicial del proyecto.

Respecto al tratamiento del dataset, se revisará el método de escalado de Ratings, puesto que lo que conlleva es a la generación de valores atípicos en la variable a predecir. Si bien se ha repartido la distribución algo mejor, sí que es cierto que los mejores Ratings ya se daban en los mejores jugadores del planeta, que juegan en los mejores equipos del planeta, por lo que el escalado actual no ha hecho sino exacerbar más la diferencia entre esos jugadores y el resto.

Se hará una revisión de la fuente de la que obtenemos las puntuaciones de equipo para revisar posibles anomalías en equipos, pero se pone como opción utilizar otros medidores para medir la fortaleza de un equipo, puesto que por motivos económicos la fuente cesó su actividad el 31 de marzo de 2020, por lo que dificulta la ampliación futura del dataset para un posible TFM.

Por último, destacar el frente que se abre si se consigue ampliar el dataset de crear un sistema que de forma automática sea capaz de sugerir fichajes. Se plantea el uso de entrenar modelos única y exclusivamente con respecto a fichajes donde se incorporen los datos del jugador y los datos del equipo en el que va a figurar y plantear un modelo de predicción. De esta forma, el usuario podría hacer una evaluación con respecto a un grupo de jugadores potenciales, incorporar la estadísticas del club para el que esté haciendo el análisis y que el sistema ofrezca un Rating estimado de los candidatos y directamente sirva como punto de partida para hacer un ojeo más exhaustivo de las mejores recomendaciones.

Bibliografía

[1] W. Rangel, C. Ugrinowitsch, and L. Lamas, "Basketball players' versatility: Assessing the diversity of tactical roles," *International Journal of Sports Science & Coaching*, vol. 14, no. 4, pp. 552–561, 2019.

[2] V. Sarlis and C. Tjortjis, "Sports analytics—evaluation of basketball players and team performance," *Information Systems*, vol. 93, p. 101562, 2020.

[3] I. Behravan and S. M. Razavi, "A novel machine learning method for estimating football players' value in the transfer market," *Soft Computing*, vol. 25, no. 3, pp. 2499–2511, 2021.

[4] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[5] A. Yiğit, B. Samak, and T. Kaya, *Football Player Value Assessment Using Machine Learning Techniques*, pp. 289–297. 01 2020.

[6] A. T. Yigit, B. Samak, and T. Kaya, "An xgboost-lasso ensemble modeling approach to football player value assessment," *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–12, 2020.

[7] P. Singh and P. Lamba, "Influence of crowdsourcing, popularity and previous year statistics in market value estimation of football players," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 22, pp. 113–126, 02 2019.

[8] P. Rajesh, M. Alam, M. Tahernezhadi, *et al.*, "A data science approach to football team player selection," in *2020 IEEE International Conference on Electro Information Technology (EIT)*, pp. 175–183, IEEE, 2020.

[9] M. Carpita, E. Ciavolino, and P. Pasca, "Exploring and modelling team performances of the kaggle european soccer database," *Statistical Modelling*, vol. 19, no. 1, pp. 74–101, 2019.

[10] H. Eggels, R. van Elk, and M. Pechenizkiy, "Explaining soccer match outcomes with goal scoring opportunities predictive analytics," in *Proceedings of the Workshop on Machine Learning and Data Mining for Sports Analytics 2016 co-located with the 2016 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2016)*, CEUR Workshop Proceedings, CEUR-WS.org, 2016.

[11] T. Decroos and J. Davis, "Player vectors: Characterizing soccer players' playing style from match event streams," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 569–584, Springer, 2019.

[12] "Opta sports." `https://www.optasports.com/`.

[13] "Wyscout." `https://wyscout.com/`.

[14] K. Apostolou and C. Tjortjis, "Sports analytics algorithms for performance prediction," in *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pp. 1–4, IEEE, 2019.

[15] M. Carpita, E. Ciavolino, and P. Pasca, "Players' role-based performance composite indicators of soccer teams: A statistical perspective," *Social Indicators Research*, pp. 1–16, 2020.

[16] V. M. Payyappalli and J. Zhuang, "A data-driven integer programming model for soccer clubs' decision making on player transfers," *Environment Systems and Decisions*, vol. 39, no. 4, pp. 466–481, 2019.

[17] C.-Y. Chen, H.-M. Feng, and F. Ye, "Automatic particle swarm optimization clustering algorithm," vol. 13, pp. 379–387, 11 2006.

[18] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, V. Vapnik, *et al.*, "Support vector regression machines," *Advances in neural information processing systems*, vol. 9, pp. 155–161, 1997.

[19] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] T. Decroos, L. Bransen, J. Van Haaren, and J. Davis, "Actions speak louder than goals: Valuing player actions in soccer," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1851–1861, 2019.

[21] L. Bransen and J. Van Haaren, "Measuring football players' on-the-ball contributions from passes during games," in *International Workshop on Machine Learning and Data Mining for Sports Analytics*, pp. 3–15, Springer, 2018.

[22] Y. Kim, K.-H. N. Bui, and J. J. Jung, "Data-driven exploratory approach on player valuation in football transfer market," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 3, p. e5353, 2021.

[23] M. B. Kursa, A. Jankowski, and W. R. Rudnicki, "Boruta–a system for feature selection," *Fundamenta Informaticae*, vol. 101, no. 4, pp. 271–285, 2010.

[24] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[25] L. Pappalardo, P. Cintia, P. Ferragina, E. Massucco, D. Pedreschi, and F. Giannotti, "Playerank: data-driven performance evaluation and player ranking in soccer via a machine learning approach," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 5, pp. 1–27, 2019.

[26] Y. Tang, "Deep learning using linear support vector machines," *arXiv preprint arXiv:1306.0239*, 2013.

[27] "Fifa 20 complete dataset." `https://www.kaggle.com/stefanoleone992/fifa-20-complete-player-dataset`.

[28] "Football manager data." `https://www.kaggle.com/ajinkyablaze/football-manager-data`.

[29] "Fifa 21: A petition has been created to get ea sports to redo the player ratings." `https://www.givemesport.com/1598764-fifa-21-a-petition-has-been-created-to-get-ea-sports-to-redo-the-player-r`

[30] "How accurate are fifa ratings compared to real life stats?." https://www.fifplay.com/how-accurate-are-fifa-ratings-compared-to-real-life-stats/.

[31] L. Pappalardo, P. Cintia, A. Rossi, E. Massucco, P. Ferragina, D. Pedreschi, and F. Giannotti, "A

public data set of spatio-temporal match events in soccer competitions," *Scientific data*, vol. 6, no. 1, pp. 1–15, 2019.

[32] "Uefa association club coefficients." https://www.uefa.com/memberassociations/uefarankings/country/#/yr/2021. Accedido: 2021-01.

[33] A. M. López, "Latin america: countries with most soccer players abroad in 2019." https://www.statista.com/statistics/872113/latin-american-countries-most-soccer-players-abroad/, 03 2021. Accedido: 2021-03.

[34] "Fifa men's world ranking." https://es.fifa.com/fifa-world-ranking/ranking-table/men/#all. Accedido: 2021-03.

[35] "Whoscored.com." https://www.whoscored.com/.

[36] "Whoscored scraper." https://github.com/joseramon-arias/scraper-whoscored.

[37] "Selenium." https://www.selenium.dev/.

[38] "clubworldranking.com." https://www.clubworldranking.com/.

[39] "Whoscored ratings explained." https://www.whoscored.com/Explanations.

[40] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the royal statistical society: series B (statistical methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[41] "Standardscaler." https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler.

[42] "Elasticnet." https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html.

[43] "Graphext." https://www.graphext.com/.

Apéndices

A

Protocolo de búsqueda bibliográfica

Para la búsqueda de investigación previa se han realizado las siguientes búsquedas mediante el motor de búsqueda científica de SCOPUS, sobre el que hicieron dos consultas.

La primera de estas consistía en la búsqueda de aplicaciones del data science y el aprendizaje automático sobre bien el mundo de los deportes, deportes electrónicos u otros aspectos. La consulta introducida fue la siguiente:

(TITLE-ABS-KEY ("football") OR TITLE-ABS-KEY ("soccer") OR TITLE-ABS-KEY ("basketball") OR TITLE-ABS-KEY ("team performance") OR TITLE-ABS-KEY ("nfl") OR TITLE-ABS-KEY (.e-sport*") OR TITLE-ABS-KEY ("nba") OR TITLE-ABS-KEY ("premier league")) AND (TITLE-ABS-KEY ("forecasting") OR TITLE-ABS-KEY ("predictive model*") OR TITLE-ABS-KEY ("machine learning") OR TITLE-ABS-KEY ("predictive") OR TITLE-ABS-KEY ("prediction model*") OR TITLE-ABS-KEY ("prediction") OR TITLE-ABS-KEY ("data science") OR TITLE-ABS-KEY ("performance model*") OR TITLE-ABS-KEY ("data driven")) AND (TITLE-ABS-KEY ("performance") OR TITLE-ABS-KEY ("player* performance") OR TITLE-ABS-KEY ("player* market value") OR TITLE-ABS-KEY ("*sport* performance") OR TITLE-ABS-KEY (ïndividual performance"))

Esta consulta arrojó 1100 resultados. Esto es prácticamente imposible leerlo en el lapso de duración del proyecto, y la búsqueda de papers relevantes llevaría demasiado tiempo. Por lo que lo que se hizo fue solamente tomar aquellas publicaciones que se publicaran a partir de 2019, lo cual hizo la búsqueda de publicaciones más rápida.

La segunda consulta se realizó en torno a un concepto experimental que al final no figura en el proyecto, relacionado con el análisis de redes sociales, que permite describir cómo se comporta una población mediante grafos. Esta tenia la siguiente estructura y arrojó 111 resultados:

(TITLE-ABS-KEY("social network analysis") OR TITLE-ABS-KEY("sna")) AND (TITLE-ABS-KEY("teams") OR TITLE-ABS-KEY("football") OR TITLE-ABS-KEY("sports") OR TITLE-ABS-KEY("basketball") OR TITLE-ABS-KEY("soccer")) AND (LIMIT-TO (PUBYEAR,2021) OR LIMIT-TO (PUBYEAR,2020))

Una vez hechas las dos búsquedas, se ha hecho un filtrado en base a títulos y abstract, puesto que muchos hacían referencia a rendimiento en ámbitos extradeportivos, psicología y análisis biomecánico

de jugadores de cara a prevención de lesiones. Debido a la gran cantidad de publicaciones, se consideró que, a excepción de alguno que cubriese otro deporte que propusiera alguna idea interesante para el proyecto, limitar la selección a estudios relacionados con el fútbol. Respecto al analisis de redes sociales se realizó la lectura de publicaciones que en su momento eran de interés, más por el tipo de datos que requiere este tipo de análisis de forma obligatoria, no se ha incluido nada al respecto a lo lardo del proyecto.

B

Descripción de columnas del dataset

En este apéndice se da una descripción de las columnas que contiene el dataset.

- Datos de jugador
  - **Nombre de jugador**(Player Name): Cadena de texto con el nombre del jugador
  - **Temporada**(Season): Cadena de texto con la temporada
  - **Equipo**(Team): Cadena de texto con el nombre del equipo
  - **Rating**(Rating): Valor que refleja el rendimiento del jugador durante la temporada
  - **Equipo siguiente**(Next Team): Cadena de texto con el nombre del equipo en el que jugará la siguiente temporada
  - **Rating siguiente**(Next Rating): Valor que refleja el rendimiento del jugador durante la temporada siguiente
  - **Edad**(Age): Edad del jugador
  - **Nacionalidad**(Nationality): Cadena de texto con la nacionalidad del jugador
  - **Posiciones**(Positions): Cadena de texto con las posiciones que ha jugado el jugador
  - **Minutos**(Minutes): Número de minutos jugados
  - **Goles**(Goals): Número de goles anotados
  - **Asistencias**(Assists): Número de asistencias realizadas
  - **Tarjetas amarillas**(Yellows): Número de tarjetas amarillas provocadas
  - **Tarjetas rojas**(Reds): Número de tarjetas rojas provocadas
  - **Disparos**(Shots): Número de disparos realizados
  - **Porcentaje de pases**(PS %): Porcentaje de pases completados
  - **Entradas**(Tackles): Número de entradas realizadas
  - **Intercepciones**(Inter): Número de intercepciones por partido
  - **Faltas realizadas**(Fouls(def)): Número de faltas realizadas por partido
  - **Fueras de juego provocados**(Off_def): Número de fueras de juego ganados (el jugador es el más cercano a su portero y un jugador rival ha recibido un pase cuando estaba entre el jugador y el portero)
  - **Regates recibidos**(DrB_def): Número de veces que el jugador ha sido superado con regates
  - **Despejes**(Clear): Número de despejes realizados
  - **Disparos detenidos**(Blocks): Número de disparos detenidos
  - **Regates realizados**(DrB_off): Número de veces que el jugador ha superado a otros jugadores mediante regates

- **Fueras de juego perdidos**(Off_off): Número de fueras de juego perdidos
- **Pérdidas de posesión**(Disp): Número de veces que el jugador ha perdido la posesión del balón del equipo
- **Pases clave**(KeyP): Número de pases que llevan a disparo de otro jugador del equipo
- **Pases realizados**(Passes): Número de pases realizados
- **Centros realizados**(Crosses): Número de centros(pases desde los laterales hacia la zona del área rival) realizados
- **Pases largos**(LongB): Número de pases largos realizados
- **Pases infiltrados**(ThrB): Número de pases infiltrados(pases a la espalda de la defensa) realizados
- **Tiros fuera del área**(OutOfBox): Número de disparos realizados fuera del área
- **Tiros en área pequeña**(SixYardBox): Número de disparos realizados dentro del área pequeña
- **Tiros en zona de penalti**(PenaltyArea): Número de disparos realizados en la zona de penalti
- **Portero**(GK): Variable de posición cuyo valor es 1 si el jugador es portero, 0 en caso contrario
- **Lateral izquierdo**(LB): Variable de posición cuyo valor es 1 si el jugador es lateral izquierdo, 0 en caso contrario
- **Lateral derecho**(RB): Variable de posición cuyo valor es 1 si el jugador es lateral derecho, 0 en caso contrario
- **Defensa central**(CB): Variable de posición cuyo valor es 1 si el jugador es defensa central, 0 en caso contrario
- **Mediocentro defensivo**(CDM): Variable de posición cuyo valor es 1 si el jugador es mediocentro defensivo, 0 en caso contrario
- **Mediocentro**(CM): Variable de posición cuyo valor es 1 si el jugador es mediocentro, 0 en caso contrario
- **Mediocentro ofensivo**(CAM): Variable de posición cuyo valor es 1 si el jugador es mediocentro ofensivo, 0 en caso contrario
- **Extremo izquierdo**(LW): Variable de posición cuyo valor es 1 si el jugador es extremo izquierdo, 0 en caso contrario
- **Extremo derecho**(RW): Variable de posición cuyo valor es 1 si el jugador es extremo derecho, 0 en caso contrario
- **Delantero centro**(FW): Variable de posición cuyo valor es 1 si el jugador es delantero centro, 0 en caso contrario

- Datos de puntuación
  - **Puntuación de equipo**(Team Ranking): Número que indica la fortaleza del equipo del jugador. A mayor puntuación, mejor es el equipo
  - **Puntuación de equipo siguiente**(Next Ranking): Número que indica la fortaleza del equipo del jugador la temporada siguiente. A mayor puntuación, mejor es el equipo.
- Datos de equipo
  - **Posición en liga**(Position): Posición donde el equipo acabó en liga
  - **Rating promedio**(Rating_team): Rating promedio de todo el equipo
  - **Goles**(Goals_team): Número de goles anotados
  - **Disparos por partido**(Shots_pg): Número de disparos por partido
  - **Tarjetas amarillas**(Yellows_team): Número de tarjetas amarillas provocadas

- **Tarjetas rojas**(Reds_team): Número de tarjetas rojas provocadas
- **Porcentaje de posesión**(Possesion %): Porcentaje de posesión media
- **Porcentaje de pases**(Pass %): Porcentaje de pases completados
- **Duelos aéreos ganados**(AerialsWon): Número de duelos aéreos ganados por partido
- **Tiros concedidos**(Shots_conceded_pg): Número de disparos realizados por rival por partido
- **Entradas por partido**(Tackles_pg): Número de entradas realizadas por partido
- **Intercepciones por partidos**(Interceptions_pg): Número de intercepciones por partido
- **Faltas realizadas por partido**(Fouls_pg): Número de faltas realizadas por partido
- **Fueras de juego provocados**(Offsides_pg): Número de fueras de juego ganados (el jugador es el más cercano a su portero y un jugador rival ha recibido un pase cuando estaba entre el jugador y el portero) por partido
- **Disparos a puerta por partido**(Shots_OT_pg): Numero de disparos a puerta por partido
- **Regates realizados por partido**(Dribbles_pg): Número de regates por partido
- **Faltas recibidas por partido**(Fouled_pg): Número de faltas recibidas por partido
- **Tiros fuera del área**(OutOfBox_team): Número de disparos realizados fuera del área
- **Tiros en área pequeña**(SixYardBox_team): Número de disparos realizados dentro del área pequeña
- **Tiros en zona de penalti**(PenaltyArea_team): Número de disparos realizados en la zona
- **Goles en jugada**(Open_Play): Número de goles marcados en jugadas
- **Goles en contraataque**(Counter_Attack): Número de goles marcados al contraataque
- **Goles a balón parado**(Set Piece): Número de goles marcados a balón parado (lanzamientos de falta)
- **Goles de penalti**(Penalty): Número de goles marcados de penalti
- **Goles en propia**(Own_Goal): Número de goles en propia favorables
- **Goles recibidos en jugada**(Open_Play_against): Número de goles recibidos en jugadas
- **Goles recibidos en contraataque**(Counter_Attack_against): Número de goles recibidos al contra-ataque
- **Goles recibidos a balón parado**(Set Piece_against): Número de goles recibidos a balón parado (lanzamientos de falta)
- **Goles recibidos de penalti**(Penalty_against): Número de goles recibidos de penalti
- **Goles marcados en propia**(Own_Goal_against): Número de goles en propia en desfavorables
- **Pases cortos**(Short_Passes_pg): Número de pases cortos por partido
- **Centros realizados por partido**(Cross_pg): Número de centros(pases desde los laterales hacia la zona del área rival) realizados por partido
- **Pases largos por partido**(Long_Balls_pg): Número de pases largos realizados por partido
- **Pases infiltrados por partido**(Through_Ball_pg): Número de pases infiltrados(pases a la espalda de la defensa) realizados por partido
- **Pases cortos en contra**(Short_Passes_pg_against): Número de pases cortos en contra por partido
- **Centros en contra realizados por partido**(Cross_pg_against): Número de centros(pases desde los laterales hacia la zona del área rival) en contra por partido
- **Pases largos en contra por partido**(Long_Balls_pg_against): Número de pases largos en contra por partido

- **Pases infiltrados en contra por partido**(Through_Ball_pg_against): Número de pases infiltrados(pases a la espalda de la defensa) en contra por partido
- **Ataques por lado izquierdo**(Left_Side): Porcentaje de ataques realizados por el lado izquierdo
- **Ataques por el medio**(Middle): Porcentaje de ataques realizados por el medio del campo
- **Ataques por lado derecho**(Right_Side): Porcentaje de ataques realizados por el lado derecho
- **Tiros por la izquierda**(Left_Side_Shots): Porcentaje de tiros realizados desde la izquierda
- **Tiros por el medio**(Middle_Shots): Porcentaje de tiros realizados desde el medio
- **Tiros por la derecha**(Right_Side_Shots): Porcentaje de tiros realizados desde la derecha
- **Tiros recibidos por la izquierda**(Left_Side_Shots): Porcentaje de tiros recibidos desde la izquierda
- **Tiros recibidos por el medio**(Middle_Shots): Porcentaje de tiros recibidos desde el medio
- **Tiros recibidos por la derecha**(Right_Side_Shots): Porcentaje de tiros recibidos desde la derecha
- **Acciones en tercio propio**(Owm_Third): Porcentaje de acciones realizadas en el tercio defensivo
- **Acciones en tercio central**(Owm_Third): Porcentaje de acciones realizadas en el tercio del medio del campo
- **Acciones en tercio rival**(Owm_Third): Porcentaje de acciones realizadas en el tercio ofensivo

C

Imágenes para análisis de equipos

En este anexo se muestran imágenes relacionadas con el análisis entre Barcelona y Paris Saint-Germain del que se extrajeron conclusiones en el capítulo 5.

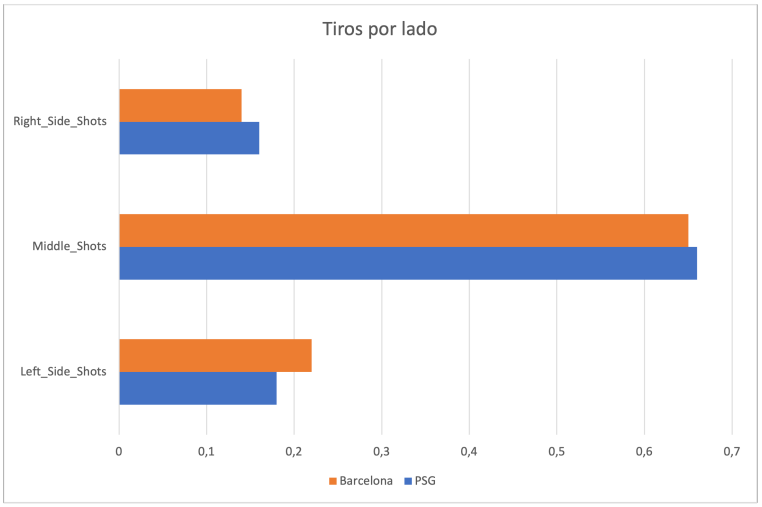


Figura C.1: Comparativa de tiros por lado entre Barcelona y PSG

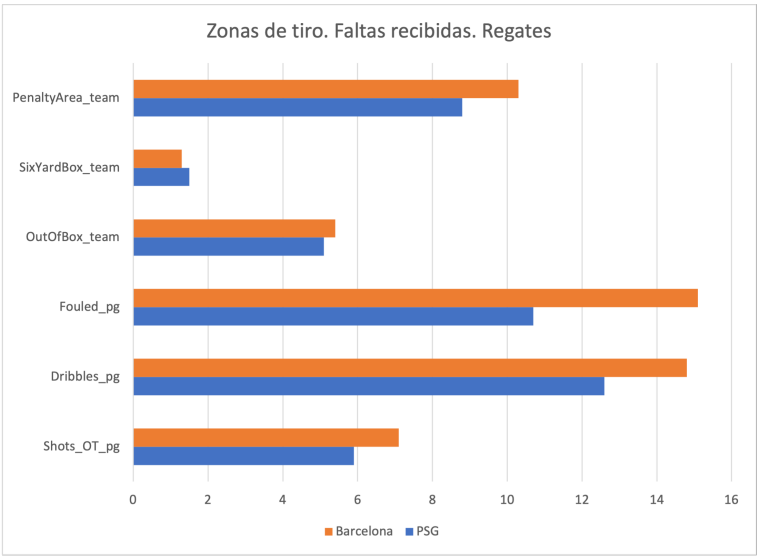


Figura C.2: Comparativa de zonas de tiro, regates, y faltas recibidas entre Barcelona y PSG

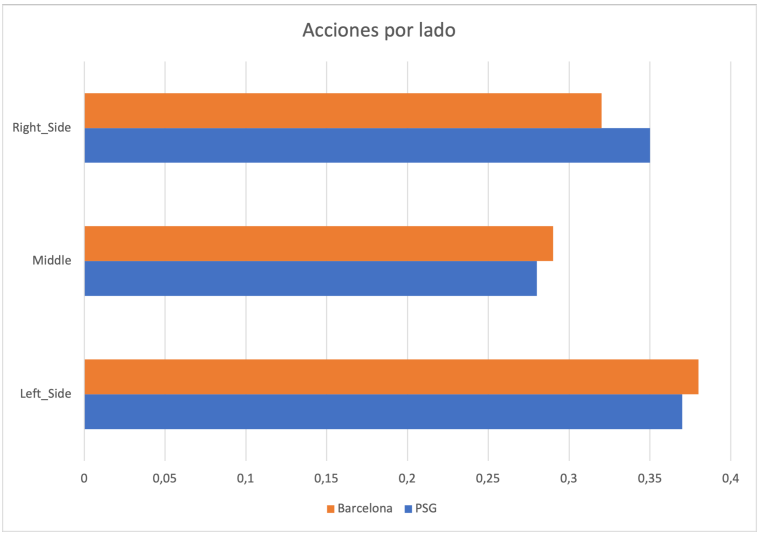


Figura C.3: Comparativa de acciones por lado entre Barcelona y PSG

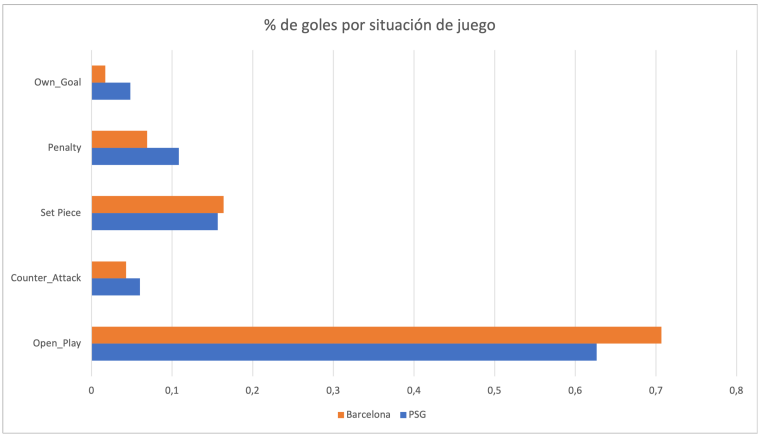


Figura C.4: Comparativa de porcentaje de goles por tipo de accion entre Barcelona y PSG


UAM
UNIVERSIDAD AUTONOMA
DE MADRID